Volume 1, No. 1

April 1984

the APPLE'S APPRENTICE

For Kids and Apples

$2.50

- Apogee and Apples, Oh Gee!
- MagiGraphics
- Bumbling Through BASIC

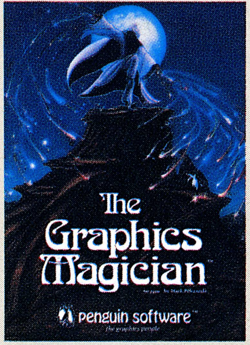
Premier Issue!

We Don't Strive for State-of-the-Art. . .
. . .We Define It.

The Complete Graphics System

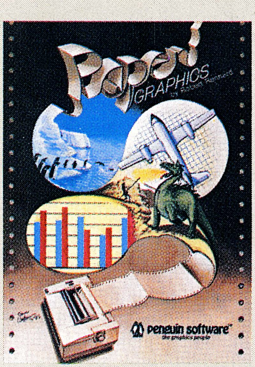
This brand-new version of our non-programmers' graphics tools includes both best-selling and highly rated products: The Complete Graphics System II and Special Effects, combined into one easy-to-use package. All the command structures have been updated so that selections are made directly by pointing at choices from a graphics screen, or options are described on convenient help screens. This version is so advanced that users will hardly need a manual at all, yet they'll have the most diverse and powerful set of graphic capabilities readily at their fingertips. And we've combined all different versions into one single package that works with joysticks, paddles, trackball, the Apple Graphics Tablet, Apple Mouse, Houston Instruments' HiPad, and the Koala Pad. Priced at $79.95, it's sure to remain the most-used graphics development tool for the Apple.

The Graphics Magician



The new version of The Graphics Magician takes all the abilities of the original version, adds to them, and simplifies their use for even the least technically-oriented programmers. Animation and picture-drawing routines from this best-seller are being used in published products from over two dozen companies, including the likes of Sierra On-Line, Sir-Tech, Milton-Bradley, Mattel, Spinnaker, Adventure International, and many others. The big news is that versions are now being released for Macintosh, Atari, IBM, and Commodore personal computers, with graphics files transferable between computers. That means that a programmer's graphics work on one computer no longer needs to be redone on other computers . . . they can just be transferred with The Graphics Magician. Retail price is $59.95 for the Apple.

Paper Graphics



Paper Graphics is a brand-new graphics screen-to-printer printing utility. As you would expect from Penguin, it's the most advanced and easy-to-use of any such utility available today. An advance, incomplete version has already received an A+ rating from Peelings II, which called it "the most complete of the graphics-dump programs reviewed to date". Besides being compatible with virtually every interface card/black and white printer combination imaginable (we challenge you to find one that it won't work with), Paper Graphics includes magnification, cropping, screen editing, labeling, framing, combination dumps of both graphics screens, and the ability to pack and unpack pictures. At $49.95, you shouldn't settle for less.

Transitions

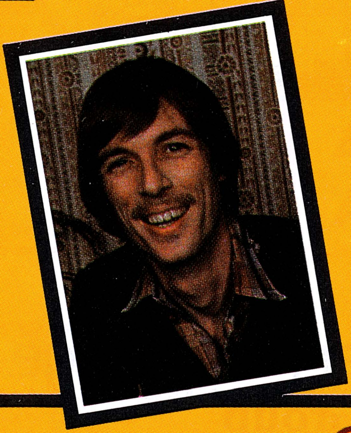
Transitions is the most advanced graphics presentation system yet on microcomputers. With it, you can easily create self-running or manually operated slide shows or presentations by combining up to eight picture disks (packed or unpacked) and 44 different transitions (screen wipes) between slides. Users can even see a graphic "catalog" of their picture disks, consisting of miniature versions of the pictures on each disk presented on the graphics screen. For a very professional-looking presentation, no other program will do. Transitions retails for $49.95, and together with The Complete Graphics System and Paper Graphics makes the most versatile set of graphics programs anyone could own for their Apple computer.

Additional Typesets and Map Pack

Two add-ons are available for The Complete Graphics System, at $19.95 each. Additional Type Sets contain over 50 extra typefaces that can be used with the text routines in CGS. Map Pack contains over 100 hi-res maps already on packed graphics screens.

penguin software™
the graphics people

830 Fourth Avenue, P.O. Box 311, Geneva, IL 60134 (312) 232-1984



southwestern data systems™

CHANGES ITS NAME TO

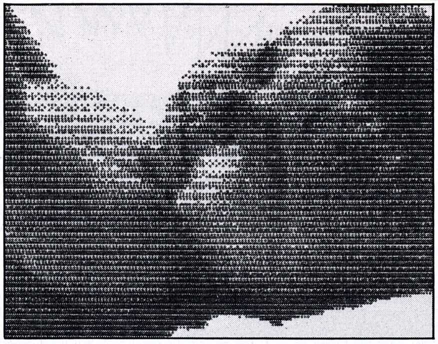
Roger Wagner Publishing, Inc.

With my name on the software, you have my personal guarantee of uncompromising quality, ease of use, and reliability in both product and support.

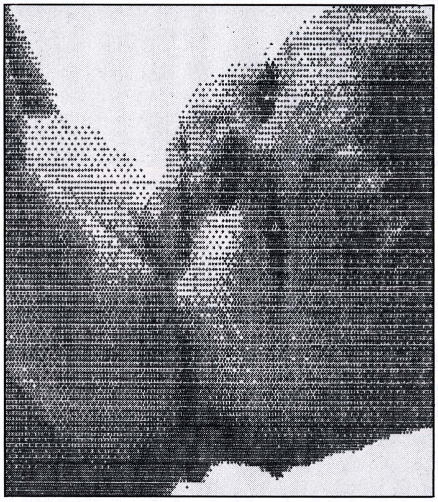
—Roger Wagner, President

First Compare, Then Demand It By Name!

"THE PRINTOGRAPHER"



OURS. . . "THE PRINTOGRAPHER"



THEIRS. . . "ZOOM GRAFIX"

Zoom Grafix is a trademark of Phoenix Software, Inc.
Apple is a trademark of Apple Computer, Inc.

If you have a printer and a printer card, Printographer is all you need to print any picture from your Apple's screen. No need for expensive graphics printer cards or other devices, Printographer simply offers the most features, and the best value. Compare for yourself. With Printographer, it's easy to see the difference.

FEATURES	THEIRS	**OURS**
Hi-Res and Lo-Res Printing	No	Yes
Reg or Inverse Inking	Yes	Yes
Picture Positioning	Yes	Yes
Magnification	Yes	Yes
Pause Printer	Yes	Yes
Color Printing	No	Yes
Add Text to Picture	No	Yes
Install in your Program	No	Yes
Completely Visual Editing	No	Yes
Oval Editing	No	Yes
Diamond Editing	No	Yes
Auto Picture Save	No	Yes
Compressed Pic Load/Save	No	Yes
Printers Supported	33	42
Interface Cards	34	34
Copyable	No	Yes
Backups	$10.00	Yes
PRICE:	**$49.95**	**$39.95**

Printographer supports virtually ALL printers including Apple Image Writer and Dot Matrix, C Itoh, Epson, Gemini, IDS, NEC, Okidata, Diablo, Qume, and Color printers such as Prism, Transtar, and C Itoh 8510 SP. Remember, the proof is in the pricture!

Requires 48K Apple II, II+, IIe, or Apple III in Emulation Mode. Order from your local Dealer. If they don't have it, we'll ship it to them within 24 hours.

Comparisons as of 3/1/84, based on Zoom Grafix (Second Edition) and Printographer (Version 4.3). Pictures shown are actual size, single magnification, and were produced on an Epson MX80 Printer.

Roger Wagner™
PUBLISHING, INC.

10761 Woodside Avenue • Suite E • P.O. Box 582 • Santee, CA 92071 • 619/562-3670

Call-A.P.P.L.E.™

Official Publication Of

Apple PugetSound Program Library Exchange
Worlds Largest Apple User Group

A.P.P.L.E. PROVIDES THE *MOST COMPREHENSIVE* USER SUPPORT SERVICES

INCREASE SALES!!!

- THE INTERNATIONAL MAGAZINE
  CALL—A.P.P.L.E.
- TROUBLESHOOTING 10 HOURS-A-DAY 7 DAYS
  HOTLINE
- MAJOR DISCOUNTS ON
  HARDWARE & SOFTWARE
- DISCOUNT SUBSCRIPTIONS TO
  SOURCE DATABASES
- **24 HR. BULLETIN BOARD**
- FOR THOSE IN PUGETSOUND COUNTRY
  - ✓ CLASSES & SPECIAL INTEREST GROUPS
  - ✓ MONTHLY MEETINGS
  - ✓ THE APPLE GENERAL STORE

COMPUTER RETAILERS & PERIODICAL DEALERS. . .
STOCK, SELL & DISPLAY
Call-A.P.P.L.E.™
Ask About "IN DEPTH" Series
CONTACT ARLISS SWORD (206) 872-2245

JOIN A.P.P.L.E. NOW & RECEIVE 10 FREE DISKS

$25 PER YEAR + $26 FIRST YEAR DUES DELIVERS A FULL MEMBERSHIP WITH ALL PRIVILEGES.
SUBSCRIPTION ONLY, $21

YES I WANT TO JOIN. . . A.P.P.L.E.

☐ MEMBERSHIP $51 FOR THE FIRST YEAR
(\$25 One-Time Application Fee + \$26 First Year Dues)

☐ SUBSCRIPTION ONLY $21 PER YEAR

Call Toll Free. . . . 1-800-426-3667

Tell operator you are joining through SOFTALK for you're FREE DISKS.

MasterCard # __________

VISA # __________

NAME __________ PHONE __________

ADDRESS __________ Exp. Date __________

CITY __________ STATE __________ ZIP __________ Exp. Date __________

Mail This Coupon To: A.P.P.L.E. 21246 - 68th Ave. S. KENT, WA 98032

Free Disk Special Expires 6/30/84

ADVERTISE TO OUR QUALIFIED BUYING MEMBERSHIP
John Elliott National Sales (206) 872-2245

Vol. I, No. 1 April 1984

The APPLE'S APPRENTICE

From the Tower 4
the Apprentice Wizard
timely topics to tickle teen trends . . . what's happening inside

Apogee? Yes, Apples, Oh Gee! 5
Mike Newton
the story behind the story behind the story of star trek and battlestar galactica

Assembling Teens:
The Sourceror's Apprentice 9
Mike Newton
an introduction [by one of merlin's friends] to the apple cpu

The Duncan Letters 13
Joe Holt
life in the castle: a traveling minstrel who brought tales of apples in a distant land

Spells and Potions 15
George Spelvin
cures for all ills of type apple

Ask the Wizard 15
tell me why my apple doesn't tick ok type stuff

An Intro to Low-Res Graphics:
MagiGraphics 18
Roger Wagner
this beginning "lesson" offers a simple "racer" program

Logical Variables the Illogical Way:
Bumbling Through Basic 20
Val J. Golding
if you think bumbling is bad, you may change your mind when you read this

Blast From the Past:
Phreaking Out 23
Cheshire Catalyst
among other things, famous people who built those little "blue boxes".

Back Talk 25
your chance to talk back and speak your mind on any subject

News and Reviews:
Ye Olde Game Shoppe 27
Val J. Golding
a medieval shopkeeper helps keep you up to date on current games, plus: a contest!

For New Game Players
An Introduction to Eamon Adventures 28
Michael Thyng
gentle hints on how to get started with eamon and other adventure games

For New Apple Users:
Better Begin With BASIC 29
Mike Collins
if you've never touched a computer in your life, now's the time

New Products
The Crystal Ball 31
check this spot each month for the very latest in hardware and software . . . a look at the future!

Microchips 32
Robert Cavey
sally buys a computer [almost]

Advertiser's Index

ArtSci . . . . BC
A.P.P.L.E. 2
Beagle Bros. 16,17
Datamost . . . . IBC
Emerald City Publishing, Inc. . . . 8
Hi Tech of Santa Cruz 14
Original Computer Camp 15
Overbyte 21
Penguin Software . . . . IFC
Roger Wagner Publishing, Inc. . . 1

PREMIER ISSUE!

The Apple's Apprentice is published monthly by:

Emerald City Publishing, Inc.
P.O. Box 582-AA, Santee, CA 92071,
(619) 562-7785.

Entire Contents copyright © 1984 by Emerald City Publishing, Inc.

Roger Wagner, publisher
Val J. Golding, editor
Donna Sexton, Advertising Director
Pamela Lambert, Circulation Manager
Margot and Al Tommervik, godparents (bless them!)

Subscriptions: $24/12 issues, $46/24 issues

From the Tower

the Apprentice Wizard

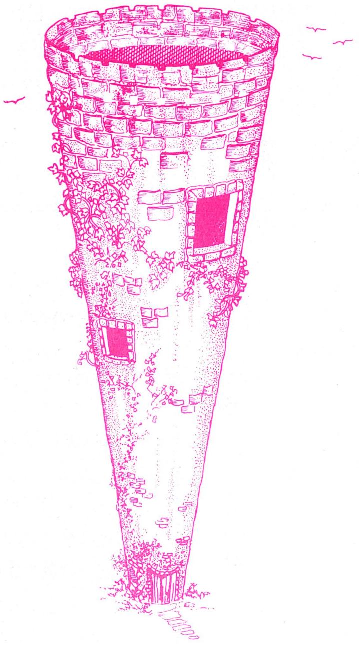
Making an Issue Of it

What an experience, putting together a new magazine the first time! You'll see it may take a while to get in the groove, get up to speed. That's why we're going to ask for your help. *The Apple's Apprentice* is going to be like nothing you've seen before, and that's because *you're* going to help create and guide your own magazine.

After you have a chance to read through this issue, write us a letter and let us know what you think. Did you like what you saw? What did you dislike? What can we add that would make it more interesting? What do you want to know about the Apple? What are your main interests... games, comic strips, graphics, programming, exploring inside? Along the same line, what do you use your Apple for? Is it your own machine, or your folks? (And speaking of your folks, we want to hear from them, too, but: *you've* gotta countersign their letter, just to let us know they're ok, ok?)

Let's see, what else? Oh, yes... (here comes another cry for help): you can be famous and see your name in print... we need some "Cheap Tricks" that will help your friends and schoolmates to learn to use the Apple better, or maybe a funny short program you have written. It doesn't even need to do anything much, as long as it does it differently (think about that for a minute), and we'll reward you for each item published with a six month extension to your subscription, or a new six month subscription for you or a friend.

Speaking of issues, this issue is dated April, 1984. For those of you who expected to see *The Apple's Apprentice* somewhat earlier, our apologies. It really does take some doing to get a new magazine to lift off. To those who expect to see a May issue, our apologies again. Our next issue will be dated June or July, and will be monthly thereafter.



Department Department

We've got a lot of neat ideas for your magazine, and you'll see many of them in this issue. Most of them will appear every month. Let's list a few:

- ✓ MICRO CHIPS
  Bob Cavey's amusing color comic about Apples
- ✓ BACK TALK
  your chance to sound off on any subject. If it's interesting, we'll print it. You'll also have a chance to reply to someone else's letter
- ✓ YE OLDE GAME SHOPPE
  news and reviews about games and gaming with an occasional contest thrown in for good measure
- ✓ SPELLS 'N' POTIONS
  open the huge oak door and find inside magical cures for ills such as how to prevent the screen from scrolling
- ✓ THE CRYSTAL BALL
  a glance through our crystal ball will pass on to you predictions of Apples in your future, games to come, the latest in peripherals, and products of interest to kids
- ✓ ASK THE WIZARD
  here's where we help *you*... If, for example, you can't figure out how to set up the ampersand (&) command?... ask the wizard

Now we come full circle: help! In this issue. be bring you as our "featured attraction," **Apogee? Yes, and Apples, Oh Gee!** Author *Mike Newton* has done a splendid job, telling you of the trauma and travail (use your dictionary) that went into the organization of a special effects production company and their later effort that resulted in pictures like *Star Wars*, *Battleship Galactica*, and *Star Trek*. Let us know if you like it. Or if you don't. Or if you'd like to hear about a 14-year old who sold an assembly language game commercially.

Deportment Department

The Apple's Apprentice is going to be a *different* magazine. Not only is it the first Apple magazine for kids, but it's the first magazine for kids where you can be the boss. Not only will we try to be responsive to your needs, we'll level with you on every word. Like the next — and very important — paragraph.

We need money. We're not a big company starting a new magazine that can afford to pour a few hundred thousand dollars into getting a project off the ground. To put out a quality magazine, for kids, and by kids, we have to sell subscriptions and advertising. We want you to look upon yourself as a co-owner. We want you to work with our staff and sell *The Apple's Apprentice*. The best salesmen in the world are those that believe in their product, and we have a product for you to shape and believe in. If you like what you see, ask your friends, your schoolmates, your teachers, to subscribe. If you really want to work at it, we'll give you a one month extension to your subscription for each new subscription you send us. Remember, for a limited time, a 12-issue introductory subscription is only $18, 25% off the regular rate.

We can predict now some changes in a typical conversation at a prom or dance such as : "Gee, Margie, how about a milkshake and Big Mac (excuse us, Merlin) after the dance?" to: "Gee, Margie, I really want to show you this new Apple magazine..." Anyway, you dig the scene... Anything you do for us will be returned to you in reading and computing delight.

So here we go... dig in and send us a report card...

Apogee? Yes, Apples, Oh Gee!

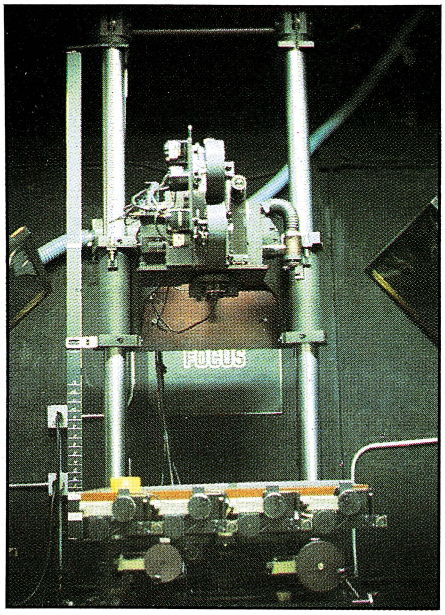
Mike Newton

How can one introduce the special effects studio that has produced some of the finest special effects shots ever seen on the wide screen? Perhaps with the same words one uses to describe their photography. Words like stunning, breathtaking, innovative and revolutionary. If these won't do, look around for one that conveys the ultimate impression: reality beyond belief.

What super-computers would it take to support this ensemble of celluloid sorcery? A *Cray One*? A *PDP-11*? Would you believe a 128-byte *Cosmac Elf*?

If you answered yes, you may be a just a little bit too gullible. You are correct, however, that a microcomputer is the workhorse — the "brains," so to speak — employed by this leading edge technological toyshop. A microcomputer called the Apple][.

You probably already know all about it, but back in 1978, few had. Along came bold Matt Beck, who proposed its use as a motion control device, a duty which, as it turned out, the Apple was well suited to. In those early days, not many people, perhaps excepting those who subscribed to *Byte* or *Kilobaud* paid much heed to micros, let alone considered a micro for the responsible task of controlling a motion picture camera.



The name of the company all of this has been leading up to, is *Apogee*. You may have also heard about Apogee before, and small wonder. In its short life span, it has produced special effects for such illustrious productions as *Star Wars*, *Battleship Galactica*, *Star Trek: the Motion Picture*, *Avalanche Express*, *Philip Marlow*, *Never Say Never Again*, and the unforgettable *Caddyshack*.

The work that is done by Apogee, and the history of how it all came about, is fascinating. To make it all simple and neatly categorized, we'll break our discussion into three parts: the history of Apogee, how the special effects systems work, and the tasks performed by the Apple itself.

A Star (Wars) is Born

In 1975, a talented young filmmaker envisioned an epic space adventure more ambitious than any science-fiction movie heretofore. This was none other than the now-revered George Lucas. He knew only he wanted to create the most realistic possible special effect for his movie, but knew of no facility capable of producing the quality desired. The solution, therefore, was to create his own.

Gary Kurtz, producer of Star Wars, happened to know of John Dykstra, a gifted cinematographer who had worked on the legendary *2001: A Space Odyssey*. The highly recommended Dykstra was hired, and *Industrial Light and Magic* was formed around him in July, 1975. John had the responsibility for assembling a group of the most talented and respected professionals in the special effects field, who would ultimately produce some of the greatest special effects ever done. Clearly, for the next couple of years, his work was cut out for him.

. . . the history of how it all came about is fascinating.

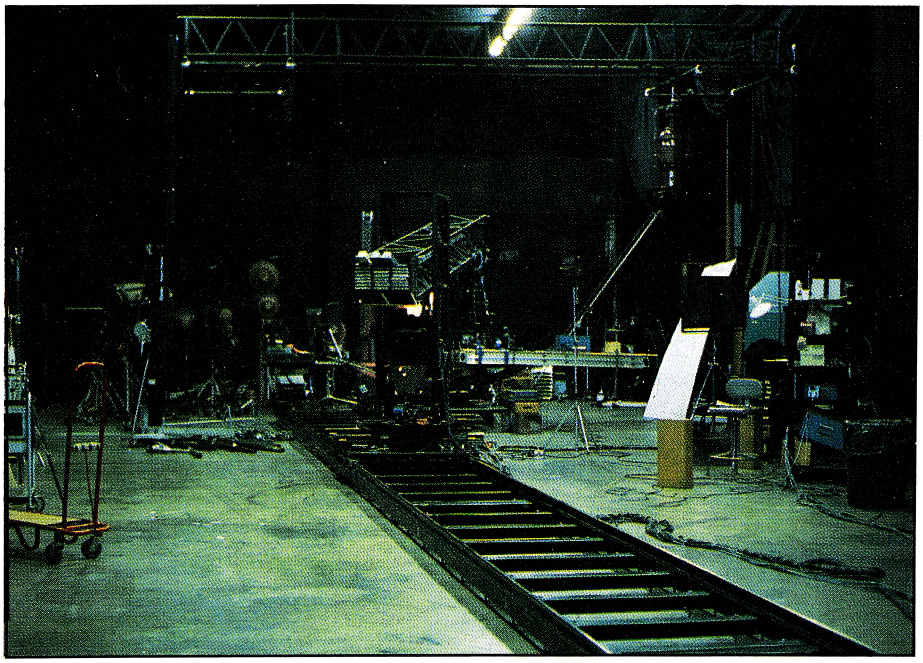
By the time Dykstra organized his team, they numbered close to one hundred, with an average age of just 27. The production crew had no time clock and no dress code. They were held together by their loyalty to Dykstra and their motivation to produce flawless special effects. (Funny, I've seen the same characteristics in programmers!)

Lucas knew only he wanted to create the most realistic possible special effects.

In order to achieve the effects Lucas desired, it was necessary to construct a device famous now as the *Dykstraflex*. It was an innovation in motion picture cameras, in that instead of requiring models to be moved past the camera, the camera itself moved in all directions, panned and tilted past the now stationary models. To incorporate the mechanical precision and memory needed by the camera brought out the genius of Al Miller, who electronically mated an elaborate system of digital clocks and multi-track recorders.

More than a year into the production, Lucas found that ILM had spent over half of their two million dollar budget, yet produced only three so-so special effects shots. After finding the facility so informal, unorganized, and unpoliced, Lucas decided to take over and personally oversee and manage the operation himself, which resulted in greatly increased tension between Dykstra and Lucas.

In the end, ILM took 22 months, including the six months needed to develop the necessary equipment, to finally finish the total of 365 special effects. When Star Wars was done, Dykstra and many of the crew left ILM to form their own independent organization, *Apogee.*



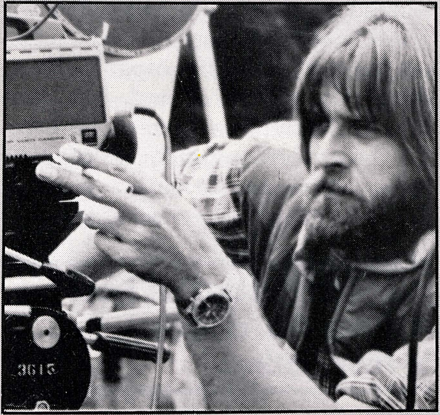
. . . it was necessary to construct a device famous now as the Dykstraflex.

Battlestar Galactica, thus far the most expensive production ever made for television, was done at Apogee, using the same equipment that had proved so effective with Star Wars. After Galactica came Star Trek: the Motion Picture, at which point Matt Beck joined the Apogee crew. Having worked previously with microcomputer hardware, Beck encouraged Apogee to utilize this technology in their motion control system.

After having examined the state of microcomputers at that time, and because of numerous difficulties using the single board *Demon* computer,

Academy Award winner John Dykstra has ventured far from his beginnings as an industrial designer to become one of the most sought-after special effects wizards in the world. Apart from his Oscar for Star Wars *and Academy Award nomination for* Star Trek, *Dykstra has garnered innumerable certificates and science-fiction awards for his contributions to filmmaking. As the Emmy award winning effects designer and producer of* Battlestar Galactica, *which enjoyed distinction as the most expensive and technologically ambitious television series ever made, Dykstra pioneered new areas of motion control photography which culminated in the creation of a camera that bears his name. The Dykstraflex Camera has become a mainstay of special effects photography, and has permitted filmmakers to capture illusions never before possible.*

As an undergraduate student, Dykstra studied architectural design while maintaining a strong interest in still photography. Shortly after college, he worked with the National Science Foundation in a psychological study of man's response to certain architectural forms, a study that sought to discover by use of miniatures, desirable formats for urban architecture. After his involvement with the N.S.F. came a brief apprenticeship with Douglass Trumbull as an industrial designer on the film Silent Running. *This film became a predecessor to the films that would dominate the late seventies and usher in the eighties.*



After completion of Silent Running, *Dykstra worked for the Ruben H. Fleet Space Theatre in San Diego on a program entitled "Voyage to the Outer Planets." This experience provided him with a background in celestial photography that would later prove most valuable.*

Following his work at the Space Theatre, Dykstra resumed his association with Douglas Trumbull and went on to produce amusement park rides, aircraft simulator films, and experimented extensively in three dimensional filmmaking.

It was at this time Dykstra became involved with George Lucas on Star Wars, *and what followed exceeded their greatest expectations. Not only did they produce a film among the most prosperous of all time, but exceeded the limitations of current technology in the greatest special effects extravaganza ever staged.*

Following his involvement with Lucas, came the 1978 formation of Apogee Incorporated, *which brought nine of the world's top special effects experts under one roof. Today they operate from a 30,000 square foot production facility, capable of turning out complete motion pictures. As president of Apogee, Dykstra oversees all productions and is presently involved in a variety of film projects.*

When not consumed with a hectic production schedule, he enjoys flying aircraft, racing cars and motorcycles, and playing music in his residential recording studio.

they decided to try their luck with an Apple][, which, as it turned out, was to be the key to their ultimate success. With its (then) high density disk drives and superior high resolution graphics, the Apple was soon put to use in the Star Trek scene where a light-energy beam wreaks havoc aboard the Enterprise.

The motion of the image distorter was controlled by an Apple . . .

The energy beam was actually a man holding a Xenon tube in front of himself. In order to disguise the tell-tale portions of the figure which appeared around the tube, they constructed an x-y-z rig that bent the picture around the light beam. The motion of the image distorter was controlled by an Apple using fast-running Integer Basic. Since none yet existed, Matt built a parallel interface for the Apple, and wrote a program to interpolate the positions of the beam for each frame of film.

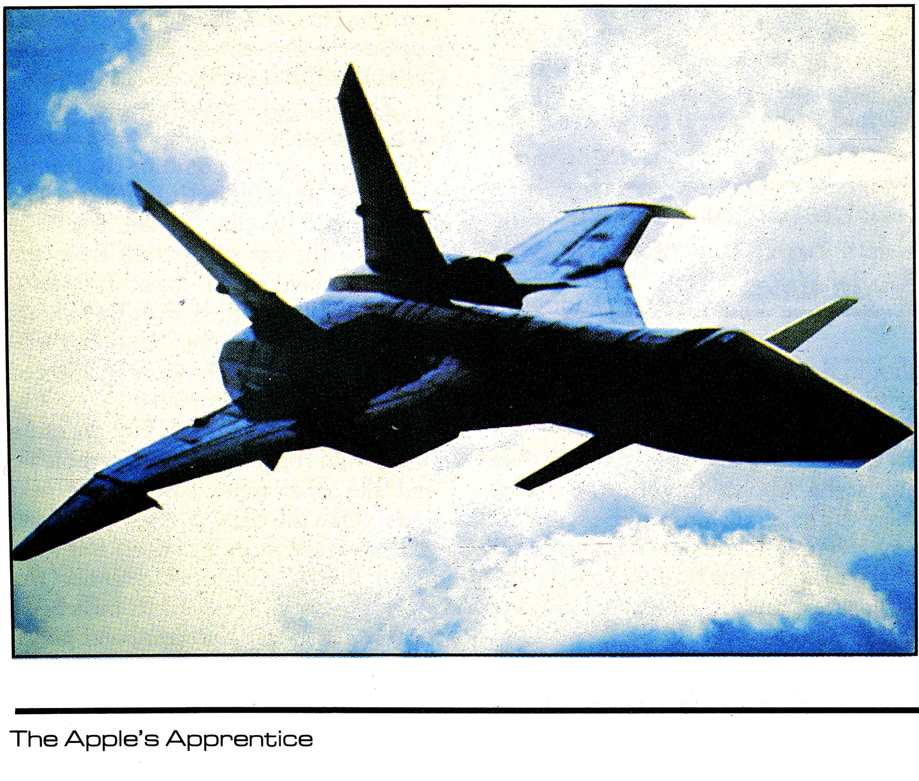
Following completion of this job, the Apple proved itself as a valuable and reliable piece of equipment. Al Miller took Beck's rudimentary interface and expanded it into a full-blown system, and from that time on, it became an integral part of the motion control system.

With the vast experience gained with Apple hardware and software, a subsidiary company, *Hollywood Hardware*, was formed. This new company has recently begun to market some of the hardware used so effectively in creating their motion picture and television special effects. Among these peripheral cards is a 16 channel A/D (analog to digital) converter with 12 bits of resolution, and a 48 line parallel interface (both of which are used in their motion control system), and the *Ultra-ROM Board.*

The latter item is of particular interest and extremely useful to Applesoft programmer types, since it includes on board *GPLE*, a BASIC edit utility which will make you wonder how you ever managed without it. (GPLE is also available on diskette from *Beagle Brothers.*) Included also on the Hollywood Hardware board are several useful ampersand routines such as a multiple byte memory search, print-using, and if/then/else, etc. All told, there is 32k of ROM space available, all transparent to the user.

Special Effects Techniques (. . . where do computers fit in?)

Most of the spectacular and dazzling special effects shots from Apogee were made possible by their computerized motion control system. Why a computer controlled camera? The answer to that lies deep in the heart of the very processes required to create special effects.



The illusion of motion with models is done with *relative* motion. In other words, instead of moving the models themselves, which tend to be somewhat immobile and cumbersome, the camera itself moves. Apogee's highly flexible camera employs twelve independent axes of motion, and can be programmed to "make passes" at models with such utter precision that only a computer could handle it. The importance of this will be obvious later.

The illusion of motion with models is done with relative motion.

A typical space scene, with space ships, stars, asteroids, lasers, etc., actually consists of many, many elements of film combined photographically. There is usually one strip of film shot for each object that appears in the picture. Sometimes as many as 30 separate rolls must be combined in an optical printer to produce the final illusion of deep space (or for that matter, near Earth).

Exactly how are these individual elements combined? The process that film makers must go through is surprisingly close to the efforts of game program authors who achieve the high quality graphics and animation that you admire.

This process is usually called *blue-screening* in the film industry, while in computer graphics circles it is known as *and/or animation.* For those of you who can relate more easily to computer concepts, graphics parallels will be used for purposes of illustration.

For example, let's say we want a ship to start off in the distance in the bottom of the frame, then zoom by over our heads. To do this with our relative motion system, we would begin by backing the camera far in front of the model, so that it appears to be very far away. The camera would have to be aimed parallel to the ground, with the model at the bottom of the frame.

The camera would begin to move toward the model, decreasing shutter speed as it advances, which creates the illusion of increasing speed. As the camera moves forward, it also begins to lower at a faster rate, all the while remaining parallel to the ground. This gives the impression that the model has suddenly gained tremendous speed and has accelerated vertically.

If we wanted to get fancy and cause the model to veer to the right, we would just slowly roll the camera to the left while the film was rolling. As you can see, these delicate camera movements are far too complex to be controlled by anything other than a computer. This illustrates the concept of relative motion with the system, but what of multiple elements? Here's where things get hairy.

. . . these delicate camera movements are far too complex to be controlled by anything other than a computer.

Suppose that we want the ship to zoom by in front of us, while at the same time, passing a large, revolving asteroid (not forgetting the starlit background).

Let's see how this might be done by a computer graphics program. The first image to be generated is the starfield, twinkling at random intervals. Next, an *and-mask*, the same size as the asteroid in the frame, would be anded to the screen to black out the area where the asteroid will appear, while leaving the remainder of the screen intact. The area where the asteroid is to be placed would be filled with zeroes, and the balance of the and-mask would contain ones. After the area occupied by the asteroid has been blanked out, we would use an *or-mask*, which would represent the asteroid in this particular frame of movement. This would be ored to the screen, resulting in the image of an asteroid in space.

The Apple's Apprentice

☐ 1 YEAR (12 issues) $24.00
(Save $6.00 over newsstand price)

☐ 2 YEARS (24 issues) $46.00
(Save $14.00 over newsstand price)

☐ Check ☐ Money Order
☐ VISA ☐ MasterCard

Card No.: ____________

Expires: ____________

Parent's Signature: ____________

Child's Name: ____________

Street: ____________

City: ____________

State: ______ Zip: ______

Emerald City Publishing
P.O. Box 582-AA • Santee, CA 92071
Telephone: 619-562-7785

The anding and oring is important. . .

The anding and oring is important because we don't want any stars to show through the asteroid, which would spoil the illusion of solidity and make the asteroid appear transparent.

Now to add to our speeding ship! Remember it is constantly moving away from us, moving diagonally upward from the lower left of the screen, and constantly growing smaller, thus each and every and-mask and or-mask will differ from frame to frame.

Now we "and" the and-mask to the screen to clear the area the space ship will occupy in a given frame. The and-mask must be used at this precise spot to be effective. Then we can "or" the image of the ship for this frame at the appropriate spot, just as we did for the asteroid. We don't want the asteroid to be seen through the ship, should the ship happen to pass in front of it. Likewise, we don't want stars to twinkle through the ship.

. . . there are several ships on the screen at any given time.

Actually, this is far simpler than any scene Apogee might do. Usually, there are several ships on the screen at any given time, plus the complications of laser fire, vapor trails, space debris, explosions, and other unnatural space phenomena.

There are several important things to take note of. First, the elements are anded or ored sequentially from the furthest back to the closest in front. This brings to mind sprites, with their priority levels and multi-planed 2½ dimensional look.

Secondly, these objects must be placed in precisely the correct locations on the screen, and at the proper speed, so as to appear synchronized and harmonized with the rest of the shot. That's where the Dykstraflex proves itself.

Now that you understand how the scene can be created with computer graphics, let's see how the photographic process might be applied to the same set of circumstances.

The blue screening process is what makes it all possible. The blue screen is a backdrop which emits a pure light frequency. This special light wavelength leaves no image or color on the film itself. Instead, the camera has a special beam splitter in it whose purpose is to separate the images. One image is transfered to regular film, while the other develops on a black and white film that produces an inverse silhouette, called a *traveling matte*. This traveling matte is then used as the and-mask, while the regular color film becomes the or-mask.

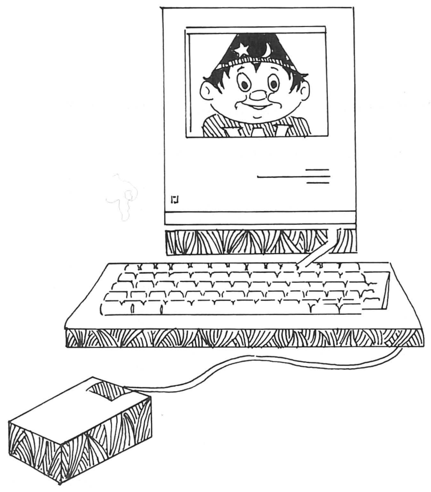
. . . the photographic process can be applied to the same set of circumstances.

These images are processed by the optical department, layer by layer. The device used to combine the frames of film is called an optical printer. The background is first anded with the matte of the object furthest from the camera, then its or-mask is photographically combined with the image. This process continues until all elements of the scene have been merged in, and the shot completed.

So far we have made little mention of the vital role played by the Apple][, and alas, that must wait for the next issue. (Like the daily comics, you don't get the whole story at once!) But keep tuned to the *Apple's Apprentice* frequency; you won't want to miss hearing about the part *your* computer plays.

Assembling Teens:

The Sourceror's Apprentice



Mike Newton

It was not too long ago, and not too far away, but the magic has been forever engraved in the Chronicles of Time. Woz was a very young wizard, but his youth did not keep him from creating magic so wonderful and so different that many of the older wizards were insanely jealous. Woz wasn't so selfish with his wizardry, though; he wanted to share it with the rest of the world. His Apple was given to many, so that they too might learn of its wonders and magic. Now it is time for you as well to learn of its magic, and become the Sourceror's Apprentice.

Any magician will tell you that the most important part of magic is learning the magic words and how to use them properly. As the Sourceror's Apprentice, you will learn how to speak to the circuits that are at the heart of the Apple] [plus and Apple //e computers, and become a wizard yourself at programming in assembly language.

Programs are written with . . . an editor/assembler.

"What is assembly language and why should I use it?" you've probably asked yourself many times. Maybe you already know there is a *microprocessor* within the computer, called the *6502.* Microprocessors are the real brains and "intelligence" of the computer, and are also called CPU's or MPU's. They "eat" electricity, "think" in binary and "talk" to other parts of the computer through wires and circuits, and *registers* or *memory cells.*

You've probably seen programs in BASIC before. It looks pretty simple. In fact, you may be tired of it already because it sometimes seems so slow. At the same time, you probably wonder what assembly language *looks* like. Before we get into that subject, let's first talk about *machine language.* Machine language is the only thing that the CPU *really* understands. Machine language looks just like an endless stream of numbers, but to the CPU, the numbers contain information that tells it what it's supposed to do, and what numbers it is supposed to use to perform calculations. It's not easy to understand a program in its machine language form, not even for experienced wizards.

. . . there is a microprocessor within the computer . . .

So how do you program in machine language? You usually don't. The preferred (and simpler) language to program in is called assembly language. Programs are written with what is called an *editor/assembler,* using using ordinary words (something like BASIC), and the editor/assembler turns all those words and text into machine language numbers. *Merlin* is an example of a good editor/assembler. Using an editor/assembler, you will see how much easier it is to write in assembly language than straight machine language.

Why would you want to program in assembly language anyway? Well, since the CPU directly understands it, it is much, much faster than BASIC. most games with graphics, for example, wouldn't look good in BASIC, because they would be too slow. Besides just being super fast, assembly is often shorter and smaller than BASIC. Not only that, but in BASIC, you are limited to using the existing "commands," whereas in assembly, you can do, the way *you* want to do it, anything the computer can do.

There are a few more reasons that make it fun to program in assembly language. For one, you are as "close" to the hardware as you can be in software. It often fills you with a feeling of awe to see your program playing directly with the innards of the computer at speeds so fast it makes you dizzy. Now that's magic for you!

What is inside the 6502? If you take the top off your Apple so that you can look in and see the *motherboard,* the biggest, black chip you will find (labeled "6502" in white writing) is the 6502 itself. Inside the 6502 are several important electronic control and memory devices, but rather than confuse you with difficult words and explanations about the 6502, I will compare it to the lower dungeons I run. There are so many similarities that you can learn about the 6502 through my dungeon comparison.

Down in the dungeon there are a bunch of strange beasts called *Hexels.* What's so strange about them? Well, instead of having ten fingers like we do, Hexels have 16 fingers, and that's just one of their strange features.

They're funny little creatures who never have much to do, so to keep them out of trouble, I try and find things to make them productive. What do they do down there? They run the Post Office department for the whole Castle.

The drawing I made for you,

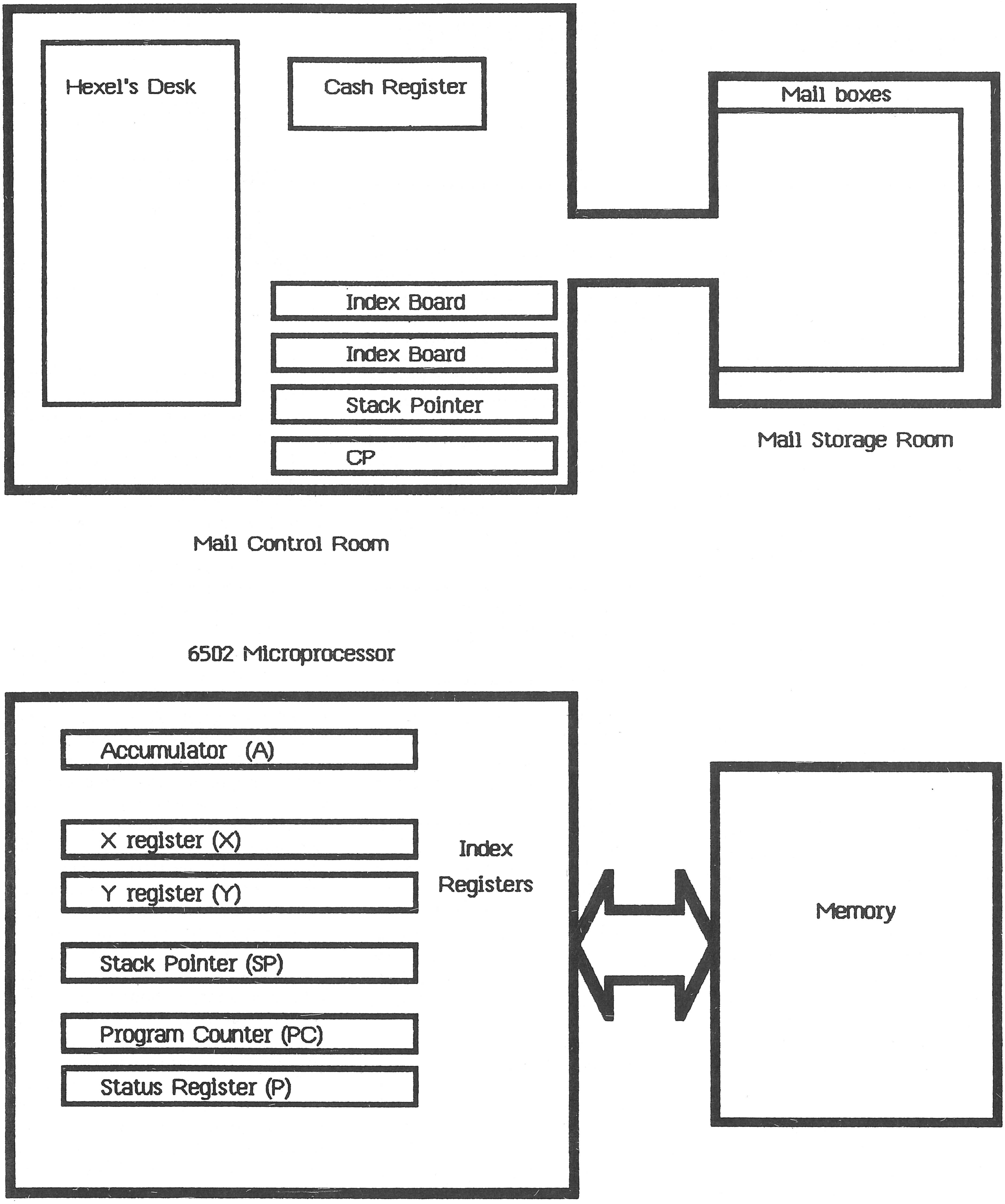

Hexel's Desk
Cash Register
Mail boxes
Index Board
Index Board
Stack Pointer
CP
Mail Storage Room
Mail Control Room
6502 Microprocessor
Accumulator (A)
X register (X)
Y register (Y)
Index
Registers
Stack Pointer (SP)
Program Counter (PC)
Status Register (P)
Memory

FIGURE 1

which I'll call *figure 1,* is an illustration of the mail control room and of the 6502. If you ask me, they look very similar. We'll talk first about the mail control rooms. You can see that there is one main control room, and that there is another room adjoining. This room is where all the mail is actually stored, in little mail boxes on the walls. We'll call this the storage room.

There is always a Hexel in the control room doing the mail. What does he do? It seems that since we do live in a royal castle, he often needs to read the mail through to make sure that no important information leaks out. For the same reason, he sometimes needs to change what is written in the letters.

Since Hexels are not very smart, they always need to be told what to do. When this Hexel comes in every morning, I leave a piece of paper telling him which mail box to get his instructions from. He goes to that mail box, and instead of taking out a letter, he finds my instructions. So you see, the same storage boxes often contain both letters and instructions.

Sometimes Hexels need to calculate things like postage cost. They aren't smart enough to do it in their heads, so they use the Cash Register in the mail room. There's only one, so when they use it, they have to be very careful not to lose any important information that might be in it.

There are two small blackboards in the control room, called the Index Boards. Because they are very small, only one number at a time can be written on them. They are used to help the Hexels remember where some of the mail slots are.

There is a special blackboard called the Stack Pointer which is sometimes used to process special mail, like the things the King sends. These special kinds of letters are always kept in mail boxes that are very close to the mail room. Since these letters are very important, this board helps the Hexel remember which letter was last put on the stack of letters. The most recent letter put on the stack is always the first one to be pulled off and mailed.

A Hexel's memory is often not very good, so there is another blackboard they write on to show which mailbox they are currently working on. This holds only one number at a time, and is called the CP (for Current Postbox).

Later on, we'll examine in detail what some of their jobs are, but in the meantime you've already gotten a pretty good idea of what they do in that dark and damp area.

The 6502 has things called registers . . .

Now let's talk about the 6502, and you'll see at once what a remarkable coincidence it is that the Hexel's job and that of the 6502 are so similar. Instead of having mailboxes for storage, it has electronic memory. Not only is memory used to store numbers and data, but like the mailboxes, it also holds the instructions or *program.*

The 6502 has things called registers that are just like the chalk boards and cash register in the mail room. Registers hold numbers, one at a time. The one that is used for all math and arithmetic calculations is called the *Accumulator.*

Just as the mail room had two blackboards called indexes, the 6502 has two *Index Registers.* They are called the "X-Register" and the "Y-Register." They are used to hold data or numbers or to "point" at places in memory. They are most often used for things called *Table Look-up.* You'll hear more about that later.

The 6502 even has a *Stack Pointer* that works just like the one in the mail room. It is used to point at very special data in a special place in memory. The pointer always points at the piece of data most recently pushed on the stack and which will be pulled off first. This is called a *LIFO* structure, which stands for "last in, first out."

In the mail room, we had a "CP" to tell the Hexels which current postbox they were working on. If you reverse the initials you will get *PC.* Now the 6502 calls this the *Program Counter,* and it tells the 6502 where in memory the current program instruction is that it is executing. In other words, if there was a program that was running, and the program was in memory at location 300, the PC would contain the number 300.

There is still another special register in the 6502 called the *Processor Status Register.* It is often called just the *Status Register* or for short, the *P-Register.* This register knows almost everything that is going on in the 6502, which register is processing what kinds of numbers, and it gives this information out by hoisting *flags.* The P-Register has seven flags that are numbered from 0 to 7. For example, by checking to see if the "Z" flag is raised, you can find out that the last number processed was equal to zero. If the number is negative, then the "N" flag will equal one. This is also called being set or turned on. These are ideas we will talk about later, so don't worry about them for now.

That's pretty much what's inside the 6502. How everything works together will be easy to understand, once I tell you more about how the mail room works. You might be confused at first, and you might have to read some things several times to really understand, but who ever said being a wizard was easy? I've been at it for over two centuries now, and I still get stuck.

You must remember that being a wizard requires a lot of time and patience. You must never give up. The road may be long and hard, but the knowledge that you gain will be well worth the time and effort involved. Knowledge is power, my master always used to tell me.

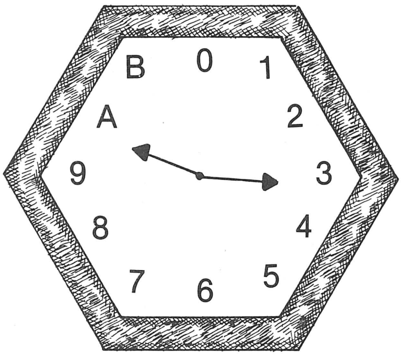
If you are really serious about learning assembly language, you need an editor/assembler, so that you can type in the example programs I will be showing you. I would recommend that you get *Merlin* from Roger Wagner Publishing. It was named after a very dear friend of mine, and he'd be quite flattered if you got it. He always was a bit of a show-off.

The Hexels are getting hungry again, so I'll having to be leaving soon. This time of night they seem to get very restless, and it's best not to leave them alone. Last time I left them by themselves, a whole colony of leprechauns disappeared, and I have a feeling I know what happened to them.

Prepare yourself for more soon, my young apprentice. Oops, I can hear those Hexels now, banging on my walls for food. They certainly do press their luck at times. Perhaps I'll have to turn one of them into a frog, just to remind them who their master is!

Anyway, before those rascals damage my dungeons, I must bid you farewell until next time . . .

Is the Hex Clock Hexed



Whoopee! Another contest! Prizes? You bet! For the first three earliest postmarked, correct solutions, you'll win 12–, 6– or 3–month subscriptions (or extensions) to *The Apple's Apprentice.* All entries must be postmarked no later than June 1st, 1984, and mailed to:

Hex Contest
The Apple's Apprentice
P.O. Box 582–AA
Santee, CA 92071

Now that you know where to send it, here's what to do: Type in Paul Raymer's Hex Clock program on this page, and RUN it. Does it run ok? Does it keep fairly good time? Good enough. Now you tell us:

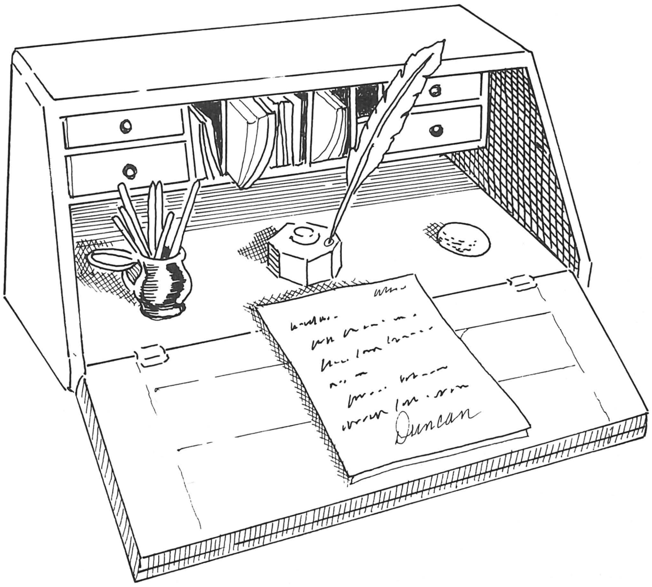
- what's wrong with the program?
- what do we do to fix it?
- did it help you learn about hex numbers?

Well, gosh, that's simple... you better get with it!

```
]LIST

100  REM

HEX CLOCK BY PAUL RAYMER

200  TEXT : HOME : CLEAR
210 A$ = "000102030405060708090A0
     B0C0D0E0F101112131415161718I
     91A1B1C1D1E1F202122232425262
     728292A2B2C2D2E2F303132333343
     53637383939A3B3C"
220 M = 1:H = 1:E$ =  CHR$ (32)
230 B$ = "       ": REM  7 SPACES
     !
240  INVERSE
250  REM

==========
BACKGROUND
==========

260  FOR X = 1 TO 15
270  FOR Y = 1 TO 40
280  VTAB X: HTAB Y: PRINT E$
290  NEXT Y: NEXT X
300  NORMAL
310  REM

======
INSETS
======

320  FOR X = 4 TO 12
330  VTAB X: HTAB 3: PRINT B$;: HTAB
     12: PRINT B$;: HTAB 23: PRINT
     B$;: HTAB 32: PRINT B$
340  NEXT X
350  REM

===========
START CLOCK
===========

360  VTAB 20: HTAB 1: INPUT "STAR
     TING HOUR (1-12) ";SH:H = 2 *
     SH + 1
370  VTAB 20: HTAB 1: INPUT "STAR
     TING MINUTE (0-59) ";SM:M =
     2 * SM - 1
380  VTAB 20: HTAB 1: INPUT "DISP
     LAY DECIMAL TIME? (Y/N) ";DE
     $: IF DE$ = "Y" THEN DE = 1
390  VTAB 20: HTAB 1: CALL  - 868

400 M = M + 2: IF M > 120 THEN H =
     H + 2:M = 1
410  IF H > 26 THEN H = 2
420  VTAB 8: HTAB 6: PRINT  MID$
     (A$,H,1);
430  VTAB 8: HTAB 15: PRINT  MID$
     (A$,H + 1,1);
440  VTAB 8: HTAB 26: PRINT  MID$
     (A$,M,1);
450  VTAB 8: HTAB 35: PRINT  MID$
     (A$,M + 1,1);
455  IF  INT (M / 2) < 10 THEN  VTAB
     23: HTAB 1: CALL  - 868: IF
     DE = 1 THEN  PRINT  INT (H /
     2);"  0"; INT (M / 2): GOTO
     470
460  VTAB 23: HTAB 1: CALL  - 868
     : IF DE = 1 THEN  PRINT  INT
     (H / 2);" "; INT (M / 2)
470  REM

=====
TIMER
=====

480  IF FL = 0 THEN  NORMAL
490  IF FL = 1 THEN  INVERSE
500  VTAB 17: HTAB 5
510  FOR D = 1 TO 31 STEP 2
520  PRINT  MID$ (A$,D,2);"  ";
530  FOR Z = 1 TO 2489: NEXT Z: REM
      (= ADJUST HERE IF NEEDED
540  IF D = 15 THEN  VTAB 18: HTAB
     5
550  NEXT D
560  NORMAL
570 FL = FL + 1: IF FL = 2 THEN F
     L = 0
580 ST =  PEEK ( - 16384): IF ST >
     127 THEN DE = DE + 1: POKE  -
     16368,0: IF DE > 1 THEN DE =
     0
590  GOTO 400
600  VTAB 23: HTAB 1: END
```

The Duncan Letters

Joe Holt



Duncan my friend!

It is always so good to hear from you. How is the young squire?

I take it from your last letter that you've begun your apprenticeship. Wonderful! May I wish you all the best of luck with your new studies? Good Luck! Keep me posted on your progress, will you?

I remember my apprenticeship... There weren't many teachers back then — what I learned, I learned on my own. You are very fortunate indeed to live in this day and time, where a bright young mind such as yours can learn and grow... I know I've told you this too many times already, but remember: Without your mind, you're nothing. Use it!

Laurelin is fine, as are the rest of the people here at the castle — little Valarel seems to have a knack for keeping our spirits high. Everyone is preparing for the coming winter and the snows she'll bring with her. I do hope it won't be as bad as the one three seasons ago.

Ah, the fire in the hearth feels so good to these cold bones of mine! Let me tell you, there is nothing quite like sitting lazily in front of a fireplace, pen in hand, while the cold winds that come off the mountains blow against the outer walls.

Oh, let me tell you about something that has caused quite a stir around here:

Yesterday, just as the sun was setting behind the distant slopes of the Far Mountains, a young man appeared outside the castle gate. His clothes gave him away immediately as being a traveling minstrel, no doubt seeking shelter within our walls from the cold. His hair was fair, and his countenance happy, despite the weather.

A minstrel he was, but the stories he told were not of dragons and hidden treasures and valiant heroes!

He spoke of a far-off land beyond the Northern Edge where, he claimed, the streets are paved with something he called "silicon" (your knowledge is more recent than mine, do you know what it is?) and the workers there have eight fingers on each hand (I'll have to ask the Sorceror if any of his Hexels have escaped). He said the people of this land often talk to fruit, or at least that's what I think he said. He did say they often carry on conversations with their apples! Can you imagine that? What's more, he said these apples talk back! Sometimes I wonder if the guards shouldn't let so many "strangers" through our gates.

He told us that these are not ordinary apples, but were specially designed to help people do all sorts of things, and that they are the creation of a young wizard named Woz. This fellow Woz, I think he ought to see the court apothecary! Really, the stories the minstrels are telling these days are getting harder and harder to believe. One should never take tales told by a man in leotards too seriously.

Whoever he was, he just left this afternoon. He said he misses his own lands, and wishes to return there as soon as he is able to get passage with a farmer or merchant going his way.

Whether these stories are true or not, they've apparently had some effect on the people here.

Everyone is anxious to see one of these "magic apples" the minstrel promised he'd bring back with him the next time he passes through the "silicon valley". That's what the Sorceror calls the place, and it seems to fit.

I can't deny that I am also a bit excited to see one of them. Being an artist, the minstrel's stories of the colorful artwork these apples can conjure up has left me curious. The minstrel has guaranteed me that anything I can put on paper, an apple can do also, only quicker and without the mess of mixing paints!

Oh, and something else I remember him saying: the pictures on these apples move! I didn't believe him either, but he promised me that what he was telling me was the absolute truth, and that he saw it happen himself. This is just too much for one as unacquainted with magic as I am to imagine.

Anyway, I'm hoping that I'll be able to use one of the apples which he brings back. I know that there are plenty of people in the Castle who hope the same, so I suppose I'll just have to wait my turn. Next time you visit me, I hope to have some beautiful artwork to show you which a "magic apple" helped me create!

I can't say I'm sorry to see him leave, however. What with all the excitement, things were beginning to get left undone. My own quarters are a mess! Now that he's gone, I hope things will return to normal, although I think that it will be a lot worse when he returns. The stories of these apples are just too good to be true! I suppose we'll all just have to wait and see.

Please write me as soon as possible — it is always so nice when the page stops by with a letter from my favorite cousin!

You are young — don't let life get old too soon! Study hard, and learn as much as you can. With knowledge, you can do anything!

your friend and cousin,

Joseph

P.S. Do your parents a favor and listen to them once in a while, all right?

WORD SEARCH

Q	W	O	R	D	F
S	E	A	R	C	H
S	T	E	C	H	I
B	B	A	K	J	L
P	D	M	O	X	Z
F	U	N	Q	G	N

WORD MATCH

___	ATMOSPHERE	1. WATER
___	BIOLOGY	2. MICROSCOPIC ORGANISM
___	BACTERIA	3. A MATHEMATICIAN
___	NEWTON	4. THE STUDY OF LIVING THINGS
___	OXYGEN	5. 14.7 POUNDS/SQ.INCH
___	LIQUID	6. COLORLESS AND ORDORLESS GAS

WORD SCRAMBLE

1. EJOYN	TERMS
2. BRSEALMC	SCRAMBLE
3. MSTER	ENJOY
4. GAAMANR	WORDS
5. YEAS	EASY
6. CLOHSO	ANAGRAM
7. WDORS	SCHOOL

TEACHER BENEFITS:
* ANY subject ANY Grade Level
* Absolutely no programming knowledge necessary
* Save 20 lessons/tests to disk
* NO GOOFS: Program traps all errors
* Money Back Guarantee

TEACHER COMMENTS:
"As a group, Word Search, Word Match and Word Scramble is the best teacher software on the market today."
"Without a doubt the most useful teacher utilities at an affordable price."
"Superior, super quality and so easy."

Special Offer
Buy all 3 programs for $60.
Save $25.
Act Now!

Word Search	$34.95	______
Word Match	$24.95	______
Word Scramble	$24.95	______
All 3	$60.00	______
Demo Disc	$ 7.00	______
Shipping	$ 3.00	______
Tax (CA only)	6%	______
Total	$	______

Visa/MC # ______

HI TECH of Santa Cruz
126 Lighthouse Ave.
Santa Cruz, CA 95060

Spells and Potions

George Spelbin

A word of introduction for this little column. What I intend to do here is to pass on little "tricks of the trade," such things as the simple and well known POKE 33,33, which narrows the text window and keeps you from getting big ugly gaps in your PRINT statements when you trace your cursor over the program line you have just LISTed.

Did you know you could POKE characters directly to the screen? Yes, indeed. Take a look at your (or your folks) Apple][Reference Manual, page 15. There is a chart there that I constantly refer to. It shows the ASCII values of Apple characters as they are displayed on the screen, inverse, flashing and normal. Note the are not the same values that Applesoft uses.

Next, turn the page and look at the chart on page 16. This is called a *memory map,* and it shows you in decimal and hex, the memory locations used for screen display. If you select location 1024 (from page 16), and POKE it with 220, guess what is displayed? Right on... the backslash (\). I'll leave the rest of the experimentation and the ideas to you. Let me know how you make out.

Just time for one more, this issue. Next time, I'll have more room. So many people have asked me how you can keep a program from being LISTed, so I'll tell you one way. Maybe at some other time I can give you more ideas.

Applesoft has what is called a *protection flag.* It is located at at 214 decimal ($D6). If your program POKEs 216 with 255 (actually any value greater than 127), any command issued from the keyboard (direct mode), such as LIST, SAVE, CALL, etc., will RUN the program. This will not work under ProDOS, and I don't know why. But look out! Make sure you have SAVEd the program before you test it, because once that "flag" is set, you can't do anything except reset or turn the power off.

Next time, I'll have a few more neat tricks to show you. I hope you'll tune in.

Ask the Wizard

Dear Wizard:

When I do a PEEK of 37 and do a VTAB later, it never seems to come out right, it's like I'm always a line off. Can you tell me why?

Sure can. It's because BASIC and the monitor count differently. Usually BASIC counts starting with one, while the monitor (because it uses hexadecimal numbers) starts with zero. Here's what really happens: On the text screen, the *cursor vertical* (decimal location 37) is a register or storage location in the monitor that keeps track of where the current screen line is. The range is from 0 (top line) to 23 (bottom line). VTAB is a BASIC command, and BASIC thinks the screen begins with line 1 and ends with line 24, so there is a conflict.

If you want to identify a screen line and VTAB to it later, do it like this: Set a variable to the screen line (X = PEEK (37), then when you want to go back to that line, VTAB X + 1. That's all there is to it!

Send your questions to "Ask the wizard," P.O. Box 582–AA, Santee, CA 92071.

TM

The Original Computer Camp

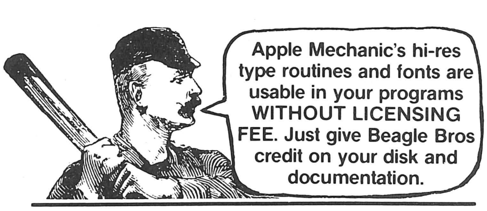
Join us for our 5th exciting season of teaching 7-15 year-olds microcomputer technology in traditional camp settings. We offer the most progressive computer curriculum available, along with complete recreational programs at a variety of ACA accredited camps. Locations include New Hampshire, Connecticut, Pennsylvania, California and Colorado. Courses in LOGO, Pascal, Robotics, Artificial Intelligence, Graphics and more. Write or call our toll-free number today!

1-800-235-6965

(inside Calif. 1-800-824-3349)

The Original Computer Camp, Inc.
559 San Ysidro Road, Dept. AA
Santa Barbara, CA 93108
(805) 969-7871

ACA
ACCREDITED CAMP


Apple Mechanic's hi-res type routines and fonts are usable in your programs WITHOUT LICENSING FEE. Just give Beagle Bros credit on your disk and documentation.

APPLE MECHANIC

HI-RES SHAPE EDITOR / TYPE FONT DISK
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #5.

SHAPE EDITOR: Keyboard-draw hi-res shapes for animation in your Applesoft programs. Access & create **proportionally-spaced hi-res Typefaces** with each character re-definable as you want. Six fonts are included on the disk. Excellent LISTable Applesoft demos show you how to animate graphics and create professional-looking Charts and Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. **Hex/Dec/Ascii displays** and input. Educational experiments included for making trick file names, restoring deleted files, changing DOS, etc.

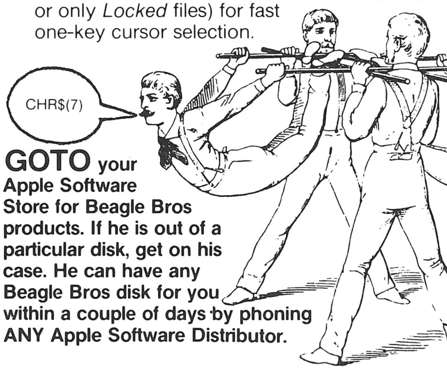
MORE: Useful music, text and hi-res tricks for your programs. Clear educational documentation.

APPLE MECHANIC
TYPEFACES
by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

26 NEW FONTS for use with Apple Mechanic programs. Many different sizes and typestyles, both ordinary and **Artistic.** Every character—from A to Z to "★" to "□"—of every typeface—from "Ace" to "Zooloo"—is re-definable to suit your needs. All typefaces are **proportionally spaced** for a more professional appearance. People *do* notice the difference!

BEAGLE MENU: Display only the file names you want from your disks (for example, only *Applesoft* or only *Locked* files) for fast one-key cursor selection.



GOTO your Apple Software Store for Beagle Bros products. If he is out of a particular disk, get on his case. He can have any Beagle Bros disk for you within a couple of days by phoning ANY Apple Software Distributor.

RUSH the following disks by First Class Mail—

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

Visa/MasterCard or COD, call **TOLL-FREE**
Orders only / ALL 50 STATES / 24 Hours a Day
1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa/MC#
to **BEAGLE BROS**, 8th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order.
Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY.

AD#8A

DOS BOSS

DISK COMMAND EDITOR
by BERT KERSEY and JACK CASSIDY

$24.00: Includes Peeks/Pokes Chart & Tip Book #2.

RENAME DOS COMMANDS & Error Messages—"Catalog" can be "Cat"; "Syntax Error" can be "Oops" or almost *anything* you want it to be.

PROTECT YOUR PROGRAMS. An unauthorized Save-attempt can produce a "Not Copyable" message, or *any* message you want. Also easy List-Prevention and other useful Apple tips and tricks. Plus one-key program-execution from catalog.

CUSTOMIZE DOS. Change the catalog Disk Volume heading to your message or title. Omit or alter catalog file codes. Fascinating documentation, tips and educational Apple experiments.

ANYONE USING YOUR DISKS (booted or not) will be using DOS the way YOU designed it.

```
10 LIST: LIST: LIST: FOR ZZ=PEEK(175)+PEEK
   (176)*256+36 TO 3072: POKE ZZ,216: NEXT
20 FOR XXX=1 TO 2: POKE-16299,0: POKE
   -16300,0: XXX=1: NEXT: REM Experiment
   with different length variable names.
```

BEAGLE BAG

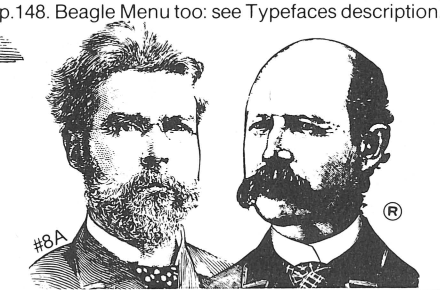
12 APPLE GAMES ON ONE DISK
by BERT KERSEY

$29.50: Includes Peeks & Pokes Chart

COMPARE BEAGLE BAG with any single-game Locked-Up disk on the market today. All 12 games are a blast, the price is a bargain, the instructions are crystal clear, *and* the disk is COPYABLE. You can even *change* the programs or list them to learn programming tricks by seeing how they work.

TWELVE GAMES from the Applesoft Ace, Bert Kersey— TextTrain, Wowzo, Magic Pack, Buzzword, Slippery Digits, and many many more...

EXCELLENT REVIEWS—See Jan-83 *Softalk*, p.148. Beagle Menu too: see Typefaces description.



Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*
(Don't Settle for Less!)

*DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

NEW!

SILICON SALAD

INCLUDING TIP DISK #2
by BERT KERSEY and MARK SIMONSEN

$24.95: Includes Peeks/Pokes AND Commands Charts

MANY MINI-UTILITIES: Disk Scanner finds bad disk sectors, **Key-Clicker** adds subtle sound as you type, **DOS-Killer** adds two tracks of space to your disks, **2-Track Cat** allows up to 210 file names per disk, **Program Splitter** makes room for hi-res pix with large Applesoft programs, **Text Imprinter** transfers text to the hi-res screen, **Onerr Tell Me** prints the appropriate error message but continues program execution, **Text Screen Formatter** converts text layouts into Print statements... plus much more Apple wizardry from the boys at Beagle Bros.

MORE TIPS ON DISK: Including fantastic programming tricks from Beagle Bros Tip Books 5, 6 and 7, plus programs from Tips/Tricks Chart #1.

TWO-LINERS TOO: From our customers around the world—and elsewhere. Little mind-blowers that will teach your old Apple some new tricks!

TIP DISK #1

100 TIP BOOK TIPS ON DISK
by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All 100 programs are LISTable and changeable for Apple experimentation.

COMMAND CHART INCLUDED: Free with each Tip Disk; an 11 x 17 poster of all Applesoft, Integer Basic & DOS Commands with Descriptions!

EARLY MODEM—

FLEX TYPE

(FORMERLY "FLEX TEXT")
VARIABLE-WIDTH HI-RES TEXT UTILITY
by MARK SIMONSEN

$29.50: Includes Peeks & Pokes Chart

PRINT VARIABLE-WIDTH TEXT on both hi-res screens with normal Applesoft commands (including HTAB 1-70). Normal, expanded & compressed text with no extra hardware. (70-column text requires a monochrome monitor, not a tv).

ADD GRAPHICS TO TEXT or add Text to hi-res graphics. Run your existing Applesoft programs under Flex Type control. Fast, easy to use, and Compatible with GPLE and Double-Take.

DOS TOOL KIT® font compatibility, or use the supplied Flex Type typefaces. Select up to 9 fonts with control-key commands. A text character editor lets you redesign any Apple text character.

FRAME-UP

FAST APPLE DISPLAY UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

PROFESSIONAL PRESENTATIONS: Turn your existing Hi-Res, Lo-Res and Text frames into attractive Apple "slide shows". *FAST* hi-res loads in 2½-seconds! Paddle or Keyboard-advance frames.

UNATTENDED SHOWS are optional, with each picture arranged and pre-programmed to display on the screen from 1 to 99 seconds. Custom **Text Screen Editor** lets you create black-and-white text "slides" and add type "live" from the keyboard during shows. Mail copies of presentations on disk to your friends and associates (or home to Mom!).

NEW!

GPLE

GLOBAL PROGRAM LINE EDITOR

by NEIL KONZEN

$49.95: Includes Peeks/Pokes Chart & Tip Book #7.

A CLASSIC APPLE PROGRAM EDITOR GPLE lets you edit Applesoft program lines *FAST* without awkward cursor-tracing and "escape editing".

INSERT & DELETE: GPLE works like a word processor for Applesoft program lines. You make changes instantly by jumping the cursor to the change point and inserting or deleting text. No need to trace to the end of a line before hitting Return.

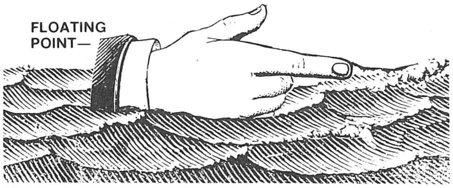
GLOBAL SEARCH & REPLACE: Find any word or variable in your programs, *FAST*. For example, find all lines containing a GOSUB, or edit or delete all lines with REM statements, or all occurrences of any variable. **Replace any variable**, word or character with any other. For example, change all X's to ABC's, or all "Horse" strings to "Cow".

80-COLUMN COMPATIBILITY: All edit & global features support **Apple IIe 80-column cards** and most 80-column cards on *any* Apple IIe, II+ or II.

DEFINABLE ESC FUNCTIONS: Define ESC plus any key to perform any task. For example, **ESC-1** can catalog drive 1, **ESC-L** can do a "HOME: LIST", **ESC-N** could type an entire subroutine... *Anything* you want, *whenever* you want.

GPLE DOS MOVER: Move DOS and GPLE to Language Card (or IIe upper 16K) for an EXTRA **10,000 Bytes** (10K) of programmable memory.

Plus APPLE TIP BOOK #7: Learn more about your Apple! Includes all new GPLE tips and tricks.


FLOATING POINT—

UTILITY CITY

21 PROGRAMMING UTILITIES

by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #3

LIST FORMATTER prints each program statement on a new line. Loops indented with printer Page Breaks. A great Applesoft program de-bugger.

MULTI-COLUMN CATALOGS, with or without sector and file codes. Organize your disk library.

INVISIBLE and trick catalog file names. Invisible functioning commands in Applesoft programs too.

MUCH MORE: 21 utilities, including auto-post Run-number & Date in programs, alphabetize/store info on disk, convert dec to hex or Int to FP, protect and append programs, dump text to printer...

LEARN PROGRAMMING: List-able programs and informative documentation. Includes Tip Book #3. Hours of good reading & Applesoft experiments.

ALPHA PLOT

HI-RES GRAPHICS/TEXT UTILITY

by BERT KERSEY and JACK CASSIDY

$39.50: Includes Peeks/Pokes Chart & Tip Book #4.

DRAW IN HI-RES on both Apple "pages" using easy keyboard commands OR paddles/joystick. Pre-view lines before plotting. Solid or mixed colors & Reverse (background-opposite) drawing. FAST one-keystroke circles, boxes & ellipses, filled or outlined. Add text for graphs & charts. All pix Save-able to disk, to be called from your Applesoft programs.

COMPRESS HI-RES DATA to 1/3 disk space (average) allowing more hi-res pictures per disk.

MANIPULATE IMAGES: Superimpose any two images, or RE-LOCATE any rectangular section of any drawing anywhere on either hi-res page.

HI-RES TYPE: Add text to your pictures with adjustable character-size and large-character color. Type *anywhere* with no Htab/Vtab limits. Type sideways too, for graphs. Includes Tip Book #4.

Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*

(Don't Settle for Less!)

* DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

BEAGLE BASIC

APPLESOFT ENHANCER

by MARK SIMONSEN

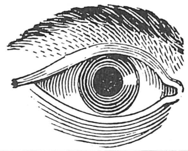
$34.95: Includes Peeks/Pokes Chart & Tip Book #6. Requires Apple IIe (OR II/II+ with RAM Card).

RENAME ANY APPLESOFT COMMAND or Error Message to anything you want. For program clarification, encryption/protection or even foreign translation. Plus add optional NEW COMMANDS:

ELSE follows If-Then statements, like this:
IF X=2 THEN PRINT "YES": *ELSE* PRINT "NO"

HSCRN reads color of any hi-res dot for collision testing. **SWAP X,Y** exchanges 2 variables' values. New **TONE** command writes music with no messy pokes & calls. **SCRL** scrolls text in *either* direction. **TXT2** lets Text Page 2 act exactly like Page 1.

PLUS: GOTO & GOSUB may precede variables, as in "GOSUB FIX" or "GOTO 4+X". Escape-mode indicated by special ESC CURSOR. Replace awkward Graphics screen-switch pokes with 1-word commands. Change ctrl-G Beep to any tone. **INVERSE REMS** too! All GPLE compatible.



```
1 FOR S=768 TO 773: READ A:
  POKE S,A: NEXT: POKE 232,0:
  POKE 233,3: DATA 1,0,4,0,5,0
2 HGR2: FOR R=0 TO 192: ROT=R:
  SCALE=96: XDRAW 1 AT 140,95:
  SCALE=30: XDRAW 1 AT 140,95:
  S=PEEK(49200): NEXT: RUN
```

PRONTO-DOS

HIGH-SPEED DOS / DOS-MOVE UTILITY

by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

TRIPLES THE SPEED of disk access and frees 10,000 bytes of extra memory by moving DOS.

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM ...	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM ...	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.

(Text Files: No Change)

Boot the Pronto disk or your updated disks, created with the normal INIT command. Compatible with all DOS Commands, GPLE, Double-Take, DOS Boss, DiskQuik and almost all unprotected programs.

MOVE DOS to your Language Card, RAM Card, or standard Apple IIe upper 16K, freeing up **10,000 EXTRA BYTES** of memory for your programs.

15 EXTRA SECTORS per disk. Catalog Free-Space displayed every time you catalog a disk.

TYPE-COMMAND ("TYPE filename") prints contents of sequential Text Files on screen or printer.

NEW!

DISKQUIK

DISK DRIVE EMULATOR

by HARRY BRUCE and GENE HITE

$29.50: Includes Peeks & Pokes Chart
Requires Apple IIe with Extended 80-column Card.

ACTS LIKE A DISK DRIVE in Slot 3, but much faster, quieter, more reliable and $350+ cheaper! Enjoy the benefits of a 2nd (or 3rd or 4th...) drive at less than 1/10th the price. Catalogs normally with "CATALOG, S3" command. Load & Save any kind of files into RAM with normal DOS commands.

SILENT AND FAST: Since no moving parts are involved, DiskQuik operates silently and at super-high speeds. See it to believe it. Your Apple IIe's Extended 80-column Card (required) can hold about half the amount of data as a 5¼" floppy disk!

MANY USES: For example, auto-load often-used files like FID etc., etc., into RAM when you boot up, so they are always available when you need them. Copy files from RAM onto disk and vice versa, just as if a disk drive were connected to slot #3.

FRIENDLY & COMPATIBLE with 80-column display, GPLE, ProntoDOS, and all normal Applesoft and DOS commands and procedures. Will not interfere with Apple IIe "Double Hi-Res" graphics.

GOTO any Software Store for Beagle Bro
If they are out of a particular disk, re ember to em to
get on the stick, and call Beagle Br are
619-296-6400 o telephon
Distribu
every
our dis
to Uncle Lou
are unprote
floppies w w
everywhe

DOUBLE-TAKE

2-WAY-SCROLL/MULTIPLE UTILITY

by MARK SIMONSEN

$34.95: Includes Peeks/Pokes AND Tips/Tricks Charts.

2-WAY SCROLLING: Listings & Catalogs scroll Up AND Down, making file names and program lines much easier to access. Change the Catalog or List scroll-direction at will, with Apple's Arrow keys.

80-COLUMN COMPATIBLE: All features support IIe and most other 80-column cards.

BETTER LIST FORMAT: Each program statement lists on a new line for *FAST* program tracing & de-bugging. Printer-compatible; any column-width.

VARIABLE-DISPLAY: Displays all of a program's strings and variables with current values.

CROSS-REFERENCE: Sorts and displays line numbers where each variable & string appears.

AUTO-LINE-NUMBER, Hex/Dec Converter, better Renumber/Append, Program Stats, Change Cursor, Space-On-Disk. GPLE/Pronto compatible.

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic ...	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA

master charge

Visa/MasterCard or COD, call **TOLL-FREE**
Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa/MC#
to **BEAGLE BROS**, 7th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping, Any-Size Order.
Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY.

CLIP COUPON OR USE SEPARATE SHEET.

AD#7A

An Intro to Low-Res Graphics:

MagiGraphics

Roger Wagner

Welcome to *MagiGraphics!* This column will appear each month in *The Apple's Apprentice* with the idea of teaching you a little about how graphics are done on an Apple computer.

You see graphics everywhere these days, whether it's in a super new Hi-Res arcade game, intriguing educational software, or even the software used by businesses.

By reading *MagiGraphics* each month, you'll not only have a better understanding of how graphics are created in these programs, but learn how to create entertaining graphics effects in your own programs!

In talking about the various subjects in this column, we'll assume that you know a little about how to write a program in Applesoft Basic, and that you know how to turn on the computer and type a program in, run it, and save it for future use.

You might be wondering about the name "MagiGraphics" for this column. It's a blending of the words "magic" and "graphics," but it's also interesting to note that the beginning of the final name then becomes *magi.* Although the word "magi" is usually associated with the three wise men in the bible story, magi actually comes from the Middle English and Latin words for sorceror! Magic is still an appropriate term though. In fact, one of the definitions for "magic" in the dictionary is "Any mysterious and overpowering quality that lends singular distinction and enchantment."

We at *Apple's Apprentice* think that the Apple computer is definitely something that has a singular distinction and enchantment to it. The computer is truly one of the most powerful things for improving your own personal creativity and productivity to come along, and the ability to use of graphics is certainly one of the more inspiring parts of that magic!

The Principles Behind Graphics

The essence of graphics is the principle of communicating information or ideas by using visual *patterns* instead of text. When you look at text, your mind uses the *words* on the page or screen to represent the things being talked about. With graphics, we use the visual image itself of an object or symbol to communicate the idea.

From the phrase "a picture is worth a thousand words," you get the impression that people have found pictures to be effective ways of communicating. One of the best recent examples of this is the success of the Macintosh and Lisa computers from Apple, and their extensive use of graphics symbols, called *icons,* to convey the ideas and instructions that ordinarily would have been done with text screen menus and volumes of instruction manuals.

Graphics images can be representations of objects we are familiar with, like houses or airplanes, or they can help us to visualize objects that have yet to be created, like the computer design systems that automobile manufacturers use to design the cars of the future. They can even be abstract patterns that have no exact meaning, they're just interesting to look at.

Graphics can include stationary images like a pie chart, or the objects can move on the screen, thus creating the effect of *animation.*

Finding the Right Spot

To create a pattern, you must have some control over where lines and shapes are drawn on the screen. To get this control, we use what is sometimes called a *coordinate system.* A coordinate system is just a big word for describing the slicing up of the screen on which we would like to draw something into a big grid, where every available spot for drawing a dot is numbered.

Usually, the screen is numbered to reflect the smallest area that can be colored in at once. This smallest area is sometimes called a *pixel.* The more pixels present in a picture, the clearer the image. The number of pixels per square inch is also referred to as the *resolution* of the picture.

Lo-Res graphics on the Apple is a short-hand term for "Low Resolution Graphics." It's called Lo-Res because the screen is only divided up into a grid that is 40 units (or pixels) wide by 40 units high. It's hard to draw anything very detailed in Lo-Res.

Hi-Res graphics is short for "High Resolution Graphics" because the screen is divided up into much smaller pixels. In Hi-Res, the screen is 280 pixels wide by 162 pixels high.

The advantages of Lo-Res are that there are more colors available, and the range of colors to deal with is smaller (40 x 40) for Lo-Res, as opposed to 280 by 162 for Hi-Res). In Hi-Res though, you can create much more detailed pictures. A pie chart or good arcade game is, for all practical purposes, impossible in Lo-Res graphics.

In both systems, each screen pixel is identified by its own pair of numbers, called a *screen coordinate.* This screen coordinate tells the computer (and you) exactly where to find or draw a given dot on the screen. The first number in the pair tells you how far over the dot is. The second number tells how far down on the screen the dot it. Drawing a dot (or pixel) on the screen is usually called *plotting* a point.

For example, in Lo-Res graphics, a dot with the screen coordinate (0,0) would be at the upper left corner of your screen. A dot with the screen coordinate (39,39) would be in the lower right corner. You'll also notice here that instead of counting from 1 to 40, the pixels are numbered from 0 to 39. There's no really great reason for this, other than it was easier for the people who designed and built the Apple computer in the first place.

Can you guess what the screen coordinates of the lower left corner and the upper right corner would be? There is a box with the answer on page 20, but think about it now for a moment to see if you've got the idea of how screen coordinates work.

Making the Computer Do Something!

To see a real example of how graphics work on your Apple, let's try doing a few experiments in Lo-Res graphics. You know, whenever you wonder whether something might work a certain way on your computer, one of the very best ways to find out is to just try it out! Experiments aren't just for scientists and science students. They're also good for helping *you* find out exactly what your computer will and won't do!

To start, type in and run this simple Applesoft program:

```
10 GR
20 FOR I=0 TO 15
30 COLOR = I
40 PLOT I,20
50 NEXT I
```

What this program does is to plot 16 boxes on the screen. The FOR-NEXT loop creates all the number values we need to get the 16 colors available in Lo-Res graphics. Line 30 sets COLOR equal to each successive value, and line 40 PLOTs the point. Notice how the screen coordinate is used in the PLOT command. Although in Lo-Res, the PLOT command actually draws a small box (because the pixel is so large), we'll generally refer to each plotted point as a 'dot'.

When I=0, the program will plot a dot at (0,20). This would correspond to a dot at the far left part of the screen, half way down. If you count the number of dots drawn on your screen by this program, you'll probably only see 15. That's because the first color plotted is *black*. Since the screen also has a black background, you don't see any change when the black dot is plotted!

Making Things Move

Now a row of dots on the screen certainly isn't all that exciting. That's why adding motion to any graphics screen makes things look so much better. The question is, how do you make the dots move?

The answer is you don't. What's that you say? You think we're playing with your mind? Well, we might be, but it's still true. You know your television or video monitor doesn't move around. And take our word for it, the letters and graphics figures you see on the screen don't *physically move* from one place to another.

What actually happens is that they're in fact erased from one position, and then *re-drawn* at a new location. If the new location is close enough to the first location (like right next to it for instance), then you're eye tells you that the letter or object has moved. Very tricky these computers! Don't feel bad though, movies work practically the same way!

To see what we mean, try this program:

```
10 GR
15 FOR J=15 TO 25
20 FOR I=0 TO 15
30 COLOR = I
40 PLOT I,J
45 COLOR = 0
47 PLOT I,J
50 NEXT I
55 NEXT J
```

Now it should look like things are moving. You should be able to tell by looking at the program though that no command like "MOVE" has been used. Instead, we've added a few lines that when combined with the original program, erase each line immediately after its drawn.

In this case, instead of drawing each row of dots at the middle of the screen, we use the counter in the FOR-NEXT loop that has 'J' as its counter to move the row down one line each time the J loop goes through one loop.

This brings us to a very important point: [If the X or Y value for a screen coordinate is changed by '1' each time the dot is plotted in a repeating loop, the dot will appear to move.]

continued on page 22

```
LIST

5  REM
RACER 1 GAME BY ROGER WAGNER

10  TEXT : HOME : GR : COLOR= 3
15  VTAB 21: HTAB 20: PRINT "HIGH
      SCORE: ";HS
20 X1 = 0:Y1 = 0:S1 = 0
30 K$ = "":D1$ = "R": REM


100  REM
MAIN SCAN LOOP


 105 KEY =  PEEK ( - 16384): IF KE
      Y > 127 THEN  POKE  - 16368,
      0:K$ =  CHR$ (KEY - 128)
 110  IF K$ =  CHR$ (27) THEN  RUN

 115  IF K$ = "I" THEN D1$ = "U"
 120  IF K$ = "M" THEN D1$ = "D"
 125  IF K$ = "J" THEN D1$ = "L"
 130  IF K$ = "K" THEN D1$ = "R"
 135  IF D1$ = "U" THEN Y1 = Y1 -
      1: IF Y1 < 0 THEN Y1 = 0
 140  IF D1$ = "D" THEN Y1 = Y1 +
      1: IF Y1 > 39 THEN Y1 = 39
145  IF D1$ = "L" THEN X1 = X1 -
     1: IF X1 < 0 THEN X1 = 0
150  IF D1$ = "R" THEN X1 = X1 +
     1: IF X1 > 39 THEN X1 = 39: REM


300  REM
DRAW PLAYER #1

305  IF  SCRN( X1,Y1) <  > 0 THEN
      COLOR= 15: PLOT X1,Y1: VTAB
     22: PRINT  CHR$ (7);"CRASH!"
     : GOTO 600
310  PLOT X1,Y1
315 S1 = S1 + 1: VTAB 21: PRINT "
     SCORE:";S1
500  GOTO 100: REM

600  REM
ANOTHER GAME?

605  IF S1 > HS THEN HS = S1
610  PRINT "ANOTHER GAME? (Y/N)";

615  HTAB 20: GET A$
620  IF A$ = "Y" THEN 10
630  IF A$ = "N" THEN  END
640  GOTO 615
```

Logical Variables the Illogical Way:

Bumbling Through Basic

Val J. Golding

Perhaps as you read our title, you thought "What is a bumbler? A bumbler is a no-where man. . ." Well, you're right and wrong, at the same time. Consider this: your first day in a new school, you find unfamiliar hallways, maybe you've just moved from another city. You've checked in with the registrar, been assigned a home room. Next task: Find the home room and go through the first-day motions of traveling from class to class.

So you go through a doorway and see somewhat forbidding doors lining each side of a hall. You're looking for 117, and on each side of the hall, you note doors labeled 101 and 162. Neither of these are close to 117, so you wander down the hall, grasping your registration form, until you reach the end. Corridors extend both left and right from the hallway, so you turn left and walk a few feet and find opposite one another, rooms 145 and 162. Wow! You're still no closer. Or are you? You've pretty much eliminated two possibilities, right? Right. So you walk back a few steps and peek into the other corridor, and the first thing that greets your eye is room 123. "Aha, I'm pretty close," you may think. And you're right. As you walk a little further, you see 121, 119 and (oh yeah!), 117! You've been bumbling, friend!

So what is bumbling, and what does it have to do with programming or the Apple? You've just done it! You've been looking for a classrooom, you checked out three hallways, and on the third try, you struck it rich; you found what you were looking for. And so it is with the BASIC programming language. To program well, you have to follow certain established ideas. But when it comes down to the nitty-gritty, you have to explore on your own. You have to often determine by trial and error whether or not something will work.

Answer to question on page 19

Lower left corner: (0,39)
Upper right corner: (39,0)

You have to determine by trial and error whether or not something will work.

There are times when you will find it expedient to ignore the rules entirely and follow your mind. Usually you will find what you are looking for. And in the meantime, you will have absorbed certain concepts: *this does this* and *this doesn't do this.* Your program may not be the best solution to a problem, but as long as it runs and does its job, that is the important thing. Later, you can (and should) go back over it and reduce things to the least common denominator, add REM lines so six months from now, you can figure out what's going on, renumber it, modularize it, in fact, rewrite it from scratch, *as long as you have a program that fits your needs* and runs correctly.

You always read about the *right* way to do things. Baloney! There are as many ways to program, and be creative in the process, as there are to drive a car. You can start easily from a stop sign, accelerate to 25 miles per hour, brake gently for the next stop, and so on, until you arrive at your destination. Or you can peel it until you hit 25 and tromp on the brakes for the next light, or you can peel off and hit your top speed, stopping for nothing until you reach your destination. All three methods usually get you where you're going (unless you run afoul of the law), but the arrival times will vary and the degree of safety will differ. The point is, there are different avenues to explore, and, at least in terms of programming, we urge you to do that. We want to give you a few examples to help you get the drift. Let's talk a bit about variables, and next time we chat, we'll drop a few hints, in our bumbling way, that will help you debug a program.

. . . there are different avenues to explore . . .

Variables are your friends. With descriptive names, they can jog your memory months later or help someone else working with your program. A variable may have a name up to 238 characters long, but Applesoft looks at only the first two characters of a variable name. Applesoft would think **RE**TRIEVE and **RE**COVER were the same variable. In writing big programs, a utility like Roger Wagner's *Apple Doc* can be a big help to index and cross-reference your variables. Note that if you get into the habit of using long (and meaningful) variable names, you gotta watch out for that old bugaboo, the *reserved word,* otherwise known as an Applesoft command. If perchance one slips into your variable name, watch out! Try this one and see what happens:

```
920 PATTERN = 45
```

There are two classes of variables (simple and subscripted) in Applesoft, and four types: real, integer (designated by %), string (designated by $), and FN. *Different classes and different types of variables may have the same name.* For instance, Applesoft would individually recognize variables named A, A%, A$, FN(A), A(0), A%(0), and A$(0). There are times where we will purposely use a real variable and an integer variable with the same name, because they relate closely to one another.

As we said a few lines back, descriptive names are good memory joggers. Typically in a FOR/NEXT loop, we will write something like

```
320  FOR I = 1 TO 45 : INPUT A(I)
     : NEXT
```

but why not write it like this:

```
320  FOR ITEM = 1 TO 45 :
     INPUT AUTHCODE(ITEM) :
     NEXT
```

Doesn't that really make a lot more sense and make the program easier to understand? Another little trick you can use — and the larger the program, the more important this becomes — is to substitute an integer variable to conserve memory. An integer (%) array will use three less bytes of memory for each element (numeric subscript) of the array. If you use integer variables, program execution speed will be slowed a trifle, but you'll probably never notice. Incidentally, the terms *array variable* and *subscripted variable* mean almost the same thing and we use them interchangeably here. An *element* is a single subscript in an array, e.g., A$(21), 21 being the element or subscript.

. . . substitute an integer variable to conserve memory.

Let's find our way back to bumbling, step by step. Did you know you can use your Apple as a lie detector? True. (ha-ha.) Of course, you may have to clue your Apple in beforehand, but if you want to check something out, Applesoft (and Integer Basic) will set a *truth flag.* To give you an idea of what we mean, type in from the keyboard:

```
A$ = "MARY"
PRINT A$ = "MARY"
```

What do you think the computer will print back? Remember, we're talking about a "truth test." If you think it will say "YES", you're wrong, but it will print "1". A truth variable can print (or return, as we like to say) only two things: the numbers "0" and "1." If the statement is true, a one will be printed, and if the statement is false, a zero will be returned. Try this:

```
PRINT A$ = "JOHN"
```

What do you think you'll see this time? 0? That's right, you're getting the idea.

Quite possibly, you're going to ask how you can use that idea in a program. Ok. Let's suppose you have an interactive (means the computer talks or responds to the user) program, and depending on whether Mary or John is using the program, different computer responses will be required. Let's look at two ways (there is always more than one way) in which we could perform a test in the program. The first, and most common method is with an IF/THEN statement like:

```
270  INPUT A$
280  IF A$ = "MARY" THEN
     TFLAG = 1
```

or you can write:

```
280  TFLAG = (A$ = "MARY")
```

This has the advantage of being just a little bit faster, and because no IF is involved, may be combined with other lines of code.

Here is where things get a little more hairy, 'cause we're going to talk

Double Byte

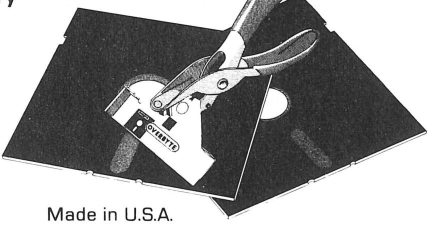
Why pay more for double-sided disks when you can use both sides of a single-sided disk. How? With OVERBYTE!

OVERBYTE is a self-aligning, heavy duty metal hold punch which doubles the capacity of a normal single-sided disk.

OVERBYTE punches a write-protect notch on the left side of a single-sided disk allowing the reverse side to be used.

OVERBYTE can be used with Apple, Rana, Franklin, Atari, Commodore, and Victor 9000 disk drives (and other compatible drives).

**Disks are expensive.
Double your memory
at half the cost!**



**OVERBYTE
ONLY $19.95**

Made in U.S.A.

Send To:

OVERBYTE™

**8621 Laurel Canyon Blvd.
Sun Valley, CA 91352
(213) 504-0309**

Please rush me OVERBYTE for $19.95

Please include $3.00 for shipping & handling (California residents add 6½% sales tax.)

NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ________ ZIP ______

PHONE (   ) ____________________

☐ MASTERCARD ☐ VISA ☐ CHECK

☐ MONEY ORDER ____________________

CREDIT CARD NUMBER ____________________

EXPIRATION DATE ____________________

SIGNATURE (for charge cards) ____________________

a little bit about *logical* variables. But before we get into that, we need to go off on a tangent, a side path, for a few minutes. Logical variables are simply extensions of truth variables, or truth variables with conditions attached.

> **. . . understanding logical variables was one of the toughest concepts we faced.**

Logical variables are modified by statements like IF, AND, OR, and NOT. When we were learning to program, understanding logical variables was one of the toughest concepts we faced. In fact we still do have to stop and think about them. Often in writing a program where logic clearly dictates the use of AND, we find that OR would be the proper form to accomplish our programming goal. But here is how we bumble our way through. If AND does not produce the expected result, then try OR. If this works, then stop and *analyze why* it works, and find the flaw in your thinking. Remember the computer is (almost) always right!

First let's dispose of IF (in its logical sense) and NOT, because not only are they the simplest of the logical variables, they are also the most mysterious. Consider two BASIC statements:

```
760  IF A = 1 THEN GOTO 1000
770  IF B = 0 THEN GOTO 1200
```

Both of the two above statements can be expressed more simply as logical variables in the form:

```
760  IF A THEN GOTO 1000
770  IF NOT B THEN GOTO 1000
```

What these two statements convey literally is *if A has a value, then*. . . and *if B has no value, then*. . .

NOT has a second function as a *toggle*. A toggle is a switch that has only two possible states, on and off. Let's say in a program you want the user to be able to change a condition at will. Using a logical variable, *the program does not need to know the current state to be able to change it.* The program statement which will do this for you is:

```
X = NOT X
```

If X = 0 then when this statement is encountered, it will be set to equal 1 or, if it now equals 1, it will be set to equal 0.

If your program needs to check whether more than one condition has been met, the IF statement can be modified, using the AND/OR combination. Generally, AND will test for and branch only when *both* conditions are true, while OR will branch to the indicated action on *either* condition being true.

```
620  IF (A AND B) THEN 1000
621  FLAG = (A AND B) :
     IF FLAG THEN 1000
630  IF (A OR B) THEN 1100
631  FLAG = (A OR B) :
     IF FLAG THEN 1100
640  IF (A OR (NOT B)) THEN 1200
651  FLAG = (A AND (NOT B) :
     IF FLAG THEN 1300
```

Now type in lines 1000, 1100, 1200 and 1300, all as STOP. When you reach one of these lines, Applesoft will print BREAK IN 1000, etc., so you'll know where your IF statement took you. Last, don't forget an INPUT or otherwise establish the values for A and B. This is bumbling at its best (or worst). Watch out for those parentheses, they'll kill 'ya every time. Strictly confidential: We almost always end up with extra sets in the programs we write!

> **Watch out for those parentheses.**

Here's a little quiz you can give yourself. Play it straight . . . write down each of the following questions and answer them, *then* RUN the above program, and find out how well you did. If you can get the hang of this, you'll find logical variables can help you out of many a tight spot. You can carry things a great deal further than we did here, and form some truly complex beasties, filled with ANDs, ORs and NOTs

> **. . . logical variables can help you out of many a tight spot.**

A equals 1 and B equals zero.
Which line or lines will be executed.
A equals -5 and B equals zero.
Which line or lines will be executed.
A equals 0 and B equals zero.
Which lines or lines will execute.
A equals 0 and B equals 256.
Which line or lines will be executed.

So AND your hearts out; we're going to knock off for a while. We'll be back to bumble some more in a future issue, and next time around, we'll talk specifically about bug extermination.

MagiGraphics

from page 19

If the dot is erased each time before it is re-drawn, the dot will look like a small object that moves. If you do not erase the old dot, then the "moving" dot will leave a trail.

You can use this "trail" to create a fun game. We call our version *RACER 1.* We have included a listing you can type in.

When the program is RUN, you'll see a line start out in the upper left corner, and head off towards the right hand part of the screen. The game is played by using the 'I,J,K and L' keys to steer the racer. If you let it run into a wall, *or even its own trail*, it will crash. You get more and more points depending on how long you can drive the racer without running into something.

The high score is printed at the bottom of the screen, so you can take turns with a friend, and see who can drive the longest without crashing.

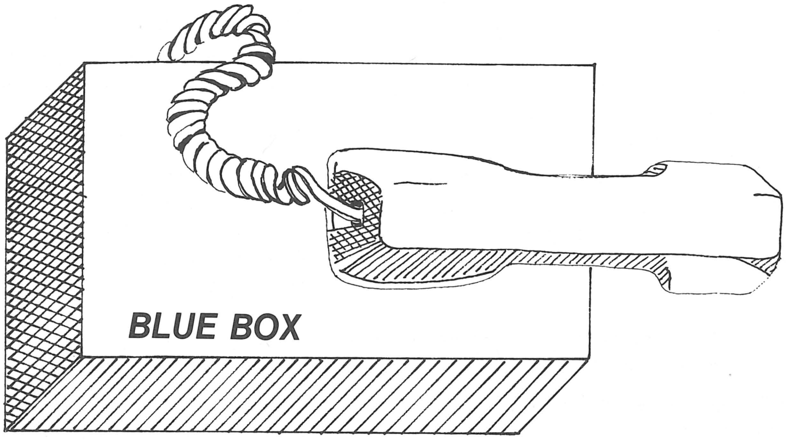
There is a logical way to make sure you get as many points as possible with the least chance of crashing. Can you figure it out?

Next time, we'll explain how this program works, and also show you a two player version that lets two people each drive their own racers at the same time on the screen. This game is very much like the "light cycles" in the movie *TRON*, and the light cycle portion of the arcade game of *TRON* .

See you next time for another session of *MagiGraphics!*

Blast From the Past:
Phreaking Out

Cheshire Catalyst



The international telephone network has been referred to as the world's largest computer. The Phone Phreaks, then, are the logical ancestors of today's modern computer hackers, since indeed, they were hacking the only available computer for anyone to hack on: The Telephone Network! If the public were made aware of how they "access" this "switching matrix," and otherwise shown the computer jargon that would pass for the simple human terms for "dialing," and "busy signal," they may possibly be made to understand that the jargon hype is a silly reason to be scared away from learning more about computers. The public "accesses" the most complex computer system on this planet every day, and almost without thinking. Certainly not thinking about it in computer-jargon, at any rate.

. . . phreaking is accessing the network in ways which were not intended . . .

Phone phreaking is accessing the phone network in ways which were not intended by designers of the telephone network. In the late 50's, Bell Labs designed the Multi-Frequency (MF) Signaling System. These MF tones were designed to directly control the Long Lines switching network. The Touch Tone® telephone dial would put out a set of tones which were completely different from the MF signalling tones. So, while you would send TT signals to your local switching office, it would then send MF tones on up the line to instruct the network as to what telephone number you wanted to be connected to.

The phone phreaks discovered the MF frequencies in the Bell System Technical Journal, a scholarly journal published by Bell Labs for network engineers so they will know what is going on with the network. When the MF signalling system was devised, it cost a lot of money to build the tone generators and decoders, even in the industrial quantities that Western Electric, supplier to the Bell System, was turning out. Not even Bell had forseen that solid state devices would drop in price so drastically that electronic hobbyists would be able to pick up the parts at Radio Shack and just throw together something that would sound to the receiving equipment like the Bell System's own equipment.

The Bell System Technical Journal, published with a blue cover, is called The Blue Book in the industry. It was no wonder, then, that the phreaks called their electronic wonder toy The *Blue Box.* It is interesting to note that the great wizard Woz was among those to develop, along with Cap'n Crunch, the first primitive blue boxes. But that is another story itself, and we'll leave that for the Woz himself to tell. To show how one of the original blue boxes worked, let's discuss how the network worked in general.

When you place a long distance call from your home, the number is captured by an "incoming register." From the first few digits, it determines that you want to place a long distance call. It then MF's the digits up the wire, and the Long Lines equipment handles it from there. When the party at the other end picks up the phone, a "supervision" signal is sent back to your central office which means "They answered, so start billing."

Just silence on the line wouldn't do . . .

When you hang up the phone, your central office stops the billing machine, and signals are sent to "dissolve" the connection.

When you've hung up, the long lines machine now has to tell the machine at the other end that the call is finished. Just silence on the line wouldn't do, because there might be a couple of teen-agers who are just on the line breathing at each other, and the Bell System wouldn't want to disconnect the call and lose the revenue. What they do do is send a tone of 2,600 Hertz (cycles per second) down the line as an indication that the line is free.

What a phone phreak did, was dial a long distance call, usually to an 800 "Toll Free" number (hotel reservations number, or an airline). The 800 numbers start billing at the other end of the call, that is, when the receiving party picks up the call, billing computers at the calling party's end are instructed to disregard 800 calls on billing tapes. Getting back to our example, however, our phone phreak will call an 800 number. While it is ringing, and before anyone can pick up the phone at the other end, the phreak will transmit a tone of 2600 Hz into the phone line. The "blue box" usually contains a speaker which is held over the mouthpiece of the telephone. The machine at the other end of the circuit "hears" its signal that the calling party has hung up, and disconnects its end of the call.

. . . the phreak will transmit a tone of 2600 Hz . . .

When the phreak stops sending the tone, the machine figures, "Oh, this trunk circuit has now been siezed by the other end, so I will now set up to receive signals (MF tones) telling me where to route the next call." The phreak then enters the sequence of tones that will route his call anywhere in the world, and the telephone company computer system snaps to an electronic "Yes Sir!," and connects the call. Now the called party picks up the phone, and the "supervision" signal is returned to the originating central office, but since that billing machine only knows that an 800 call was placed, it discards the billing information. It is (or was) as simple (?) as that. Let's face it, the phone phreak had to know as much about the network as any telephone traffic engineer. Many of the phreaks, in fact, got caught when they reported network problems they discovered.

Ah, those were the good old days. Days when phone phreaks were real phone phreaks, and not just someone trying to rip off The Phone Company (or TPC, after the 1966 movie "The President's Analyst" in which The Phone Company was foiled in its attempt to rule the world). The phreaks were genuinely interested in how the network operated. However, all good things must come to an end, and the evolving digital telephone network did away with the transmission of "analog" (sound) signals for the switching of telephone calls.

The phreaks were genuinely interested in how the network operated.

The new Common Channel Interoffice Signaling system (CCIS) is a new computerized method for The Phone Company to switch phone calls. For example, where there is a 500 pair cable between New York and Chicago, then pair number 1 would be connected to the computers at either end, and the computers would tell each other how to switch the other 499 pairs in the circuit. Besides being a faster method of communicating between the switching machines, computerized switching also did away with the "blue boxing" antics of the phone phreaks.

. . . computerized switching did away with "blue boxing" antics . . .

By that time, computers had appeared on the scene. The type of intellectual challenge that the phone network represented, was to be found in the home computers that were becoming affordable, as the phone network became more and more unexplorable.

As a case in point, John Draper of California was one of the nations best known phone phreaks after being written up in *the* definitive article on the subject in a 1971 issue of Esquire magazine. He called himself Cap'n Crunch, because he had found that a Bos'n whistle, given away free in Cap'n Crunch breakfast cereal, produced a tone of exactly 2600 Hertz! He subsequently found most of the other information he needed from Bell System publications, and by making himself obnoxious at Bell System switching offices. I've watched his interviewing technique on unsuspecting, and helpful service people, and it certainly is a wonder that he can take a person's natural inclination to be helpful, and turn it against a person's inclination to protect the proprietary nature of the information in his/her care.

Bell eventually caught him doing his nasty electronic stuff, and had him thrown in the slammer. If John hadn't been made such a folk hero by the press, it may not have gone so bad for him. On one of his sojournes into jail, (after having been caught in Pennsylvania "breaking parole") he was back in a California pokey in a work program. He was allowed out to work a job during the day, but had to be back inside the walls at night. At the time, there were no good word processing programs for the Apple computer, so John set out to write one. It was while in the work program that John wrote Easy Writer, which was the first really good word processor for the Apple.

John Draper wrote Easy Writer.

A couple of years later, IBM was looking around for software to bundle with its PC (Personal Computer) so that they would have software available when it "Hit the street." By that time, there were better packages than Easy Writer for the Apple, and other micros, but I personally think that someone at IBM had a sense of humor. Either that, or they wanted to send a shot across that someone's bow.

AT&T and IBM were taking their corners in the fight of the century.

In the late 70's, it looked as if the 80's would be the "Battle of 56th St." On the North side of Madison Avenue and 56th St in New York City, there is the world headquarters building of IBM, and on the south side of 56th St, is the new AT&T Chippendale. With the FCC unable to draw the lines between Data and Voice Communications after two inquiries to try to do so, it looked as if AT&T and IBM were taking their corners in the fight of the century. The Justice Department breakup of AT&T disrupted this fight, and the whole arena changed shape.

Meanwhile, IBM decided John Draper, the infamous Cap'n Crunch, and his new software company "Cap'n Software" should design and program the word processing package for the new IBM PC. This could be taken as a slap in the face to AT&T, if anyone bothered to look at it, but no one I know in the industry has decided to look at it this way.

So we now have a digital telephone network, and former phone phreaks now involved with digital technology. One of the early dreams when CCIS was first announced, was for a phreak to take a Gunn Diode, and make a "Gunn," or a shoulder fired rifle type of transmitter with a microcomputer inside, and "fire" a signal at the local microwave tower, which would inject a digital message saying "When my friend in Los Angeles hangs up, ring me back." The computerized switching network is able to handle messages of that type, and other services which only recently have been offered to the public, mostly because Bell couldn't determine what to charge for them.

The reason this type of thing won't work is due to the national security aspect. It seems that the Russians have been listening in to our microwave transmissions over the years. All they had to do was tune in the channels, and let the computers take over. As the MF signals were sent, the Soviet gear would determine if the "target" phone number was of interest to them. this meant, if the number being called was a military installation, or government facility, a tape recorder could be started up on that circuit. Now the Bell System and national security boys are getting together, and all the data transmissions regarding how the calls will be switched will go via under ground circuits. This way, the data being sent underground will tell the circuit switchers at the microwave sites where the call will be routed next.

As the phone phreaks get deeper and deeper into the digital universe, they are being bought up (hired) by the complex world of data communications. They've learned more by themselves, than what was being taught in colleges just a few short years ago. They are infiltrating the "Real World," as well as the computer hackers of my aquaintence, who mostly are trying to put their old "hacking" days behind them in the closet, along with the rest of the skeletons, as they penetrate further into the corporate environment, where having been "one of those nasty hackers" may not be good for their corporate health. Fortunately, there is now a movement afoot to "clear" the good name of Hacking which has recently been tarnished by the media coverage of "hacking incidents."

The phone phreaks of yesteryear are the electronic crossword puzzle solvers of the pre-computer era. Now that computers are everywhere, that kind of intellectual challenge is provided by the most complex, and involving of all mind toys, The Computer.

Cheshire Catalyst is Richard Cheshire, a new York City based Telecomputer Consultant who publishes TAP, The Hobbyist Journal for the Communications Revolution (a technological revolution only, mind you). TAP has been published for informational purposes only since 1971, and is a source of information for phone phreaks, and computer hackers. Their address is:

TAP, Room 603, 147 West 42nd
St, New York, NY 10036

This article is an exerpt from Digital Deli, a book edited by Steve Ditlea to be published by Workman Publishing.

Back Talk

Dear Val,

Being an average 16 year old with a somewhat limited computer background, I thought that I would sit down and drop you a line or two about what I would like to see in a computer magazine oriented toward the "younger generation." I think that the idea for such a magazine for the young (and the young at heart) is certainly worth whatever effort and expense there may be to bring it into fruition. Ideas concerning content, style, and frequency are probably coming to you by the dozen, however, please indulge me to give you another. Again, let me stress the point that I am not a computer expert, just one who has an interest in this new technology.

As a final thought, some of your readers may be considering careers in computers. I think that it would be nice if you, or a contributing author, would present some articles on the educational requirements necessary, and career opportunities that one could expect.

These are just a few of the thoughts that I have regarding your new magazine. I hope that they stimulate other possibilities for you. I also wish you the best of luck with the magazine, flourishing and eventful years, and a bright future.

Michelle Craft
Renton, WA

Dear Editor,

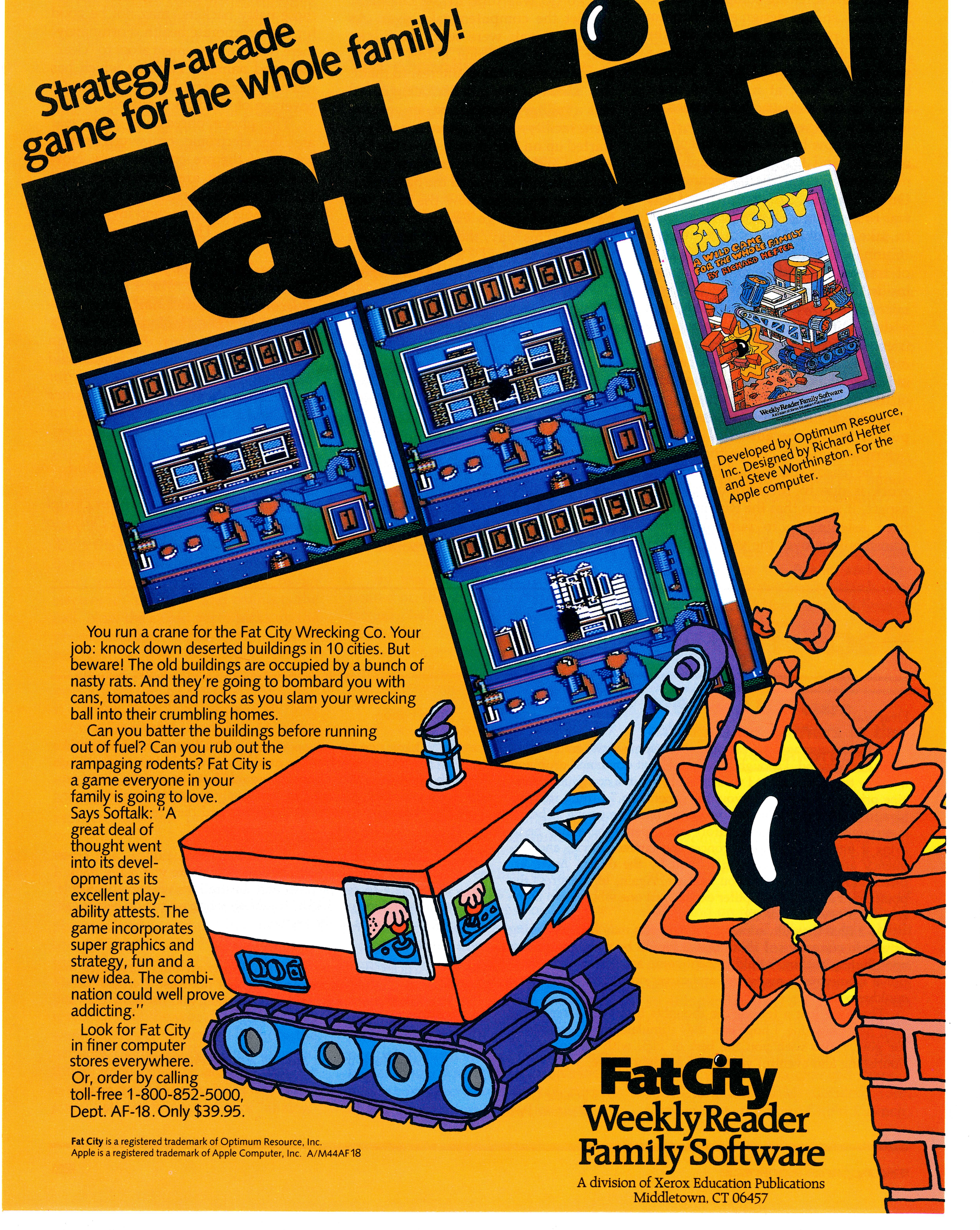
Thank you for the opportunity to tell you what I would like to see in a kids computer magazine. I would like to see things about other kids of all ages. I would like to see articles about programs written for learning, especially those that make learning fun. Of course, the latest on games and their reviews.

Thinking about games, I love puzzles, riddles, and crosswords. I would like to see articles that make you really think. I would like to see pictures and illustrations whenever possible, also something on BASIC troubleshooting we can do before calling our parents to the rescue.

Articles on many different [programming] languages and how we can use them. And last, but not least, I would mostly like to see a column on computer slang and its many uses and meanings.

Good luck to you and the *Apple's Apprentice.*

Marquisha Davis (age 9)
Van Nuys, CA


FAT CITY
A WILD GAME FOR THE WHOLE FAMILY
BY RICHARD HEFTER
Weekly Reader Family Software

Strategy-arcade game for the whole family!

Fat City

Developed by Optimum Resource, Inc. Designed by Richard Hefter and Steve Worthington. For the Apple computer.

You run a crane for the Fat City Wrecking Co. Your job: knock down deserted buildings in 10 cities. But beware! The old buildings are occupied by a bunch of nasty rats. And they're going to bombard you with cans, tomatoes and rocks as you slam your wrecking ball into their crumbling homes.

Can you batter the buildings before running out of fuel? Can you rub out the rampaging rodents? Fat City is a game everyone in your family is going to love. Says Softalk: "A great deal of thought went into its development as its excellent playability attests. The game incorporates super graphics and strategy, fun and a new idea. The combination could well prove addicting."

Look for Fat City in finer computer stores everywhere. Or, order by calling toll-free 1-800-852-5000, Dept. AF-18. Only $39.95.

Fat City
Weekly Reader
Family Software
A division of Xerox Education Publications
Middletown, CT 06457

Fat City is a registered trademark of Optimum Resource, Inc.
Apple is a registered trademark of Apple Computer, Inc. A/M44AF18

YE OLDE GAME SHOP

VAL J. GOLDING

Welcome, ye, to Ye Olde Game Shoppe. Swing wide the oaken door, my lovelies, taking care not to disturb the cobwebs, and feast thy eyes upon the goodies which lineth the walls. Wander to thy heart's content, lads and lasses, and when thou spoteth a program sufficient to curdle thy blood, feel free to place it in the magic apple and playeth away. Remember the wizard's words, my friends: "Thou breaketh, thou buyeth!" And now young ones, as the day wears on, let me direct thy attention to a special game, and to a contest thou shalt enjoy.

Fat City

Here is a game which should appeal to the destructive element in all of us. Imagine being let loose in a town doomed to be torn down and rebuilt, with a wrecking crane at your disposal (you know, the old-fashioned kind with a huge swinging ball), and no one around to stop you from climbing in the empty cab. What a ball! ('scuze the pun.) This is what *Fat City* is all about.

Visuals are great and the packaging is durable and attractive. Fat City comes with excellent documentation, decals, posters, and a warranty card for free replacement if your disk goes bad. It's $39.95 from *Weekly Reader Family Software*, 245 Long Hill Road, Middletown, CT 06457, or your dealer.

Play commences with a full fuel tank and the image of the city through the window of your cab. Keys or joystick control the motion of the crane up and down the street, and the height of the wrecking ball, which is swung with button zero. Points are scored for each portion of a building brought down, and the larger the piece the more points you get. But a word of caution: each swing of the ball uses fuel, and an attempt to bring down too large a piece will result in only a crack in the building.

One unexpected hazard is the rats. Yes, we said rats. You know rats are always the last to leave a sinking ship. Well it appears they are also the last to leave an abandoned building. In the meanwhile, they pass the time by throwing bricks and other debris at the crane. If you don't dodge, each hit costs you fuel. Ah, fuel. That's the other point we wanted to mention. There is always a fuel tank hidden behind one of the buildings, and you need to refill to be able to finish the job. Sneakily, the fuel tank is not in the same place each time you play.

When you manage to tear down an entire town, there is another waiting for your gentle touch, just ever-so-slightly tougher, up to a total of ten. The game is a joy to play, goat-getting and a whole bunch of fun. You need to spend a lot of time developing skills and technique before getting much beyond the first two or three cities.

Fat Chance!

A Contest! Here's one you can't pass up. We've spent many hours playing Fat City, but never made it passed round eight. Almost into nine, but not quite. (Those dawgone fuel tanks!) So here's a chance to win a free game or get a one year sub to *The Apple's Apprentice*. To the first reader who sends us a Fat City diskette with a score of 100,000 or more and shows round 10 completed, we'll award a free Weekly Reader Family Software game, and for the second diskette we receive, we'll give you a one year subscription or extension to *The Apple's Apprentice*. Each entry must contain the following:

- Your name, address and age
- A stamped, self-addressed envelope with sufficient postage to return your diskette, otherwise diskettes *can not* be returned
- Address your entry to:

  Fat Chance
  Emerald City Publishing, Inc.
  P.O. Box 582-AA
  Santee, CA 92071

Sounds almost too good to be true, doesn't it! But that's right, all you have to do is have fun and mail your disk to us. What more can you ask? (We'll probably hear about that!)

A Reading Adventure

For you adventure game players, we've found a new book to help you on your way through the caverns and rooms of games like *Time Zone* or *The Wizard and the Princess*. We couldn't believe it when we saw it — here are room maps and clues galore, advice and tips on playing each of the 14 popular games covered, all in large easy-to-read type, and many illustrations.

It's 167 pages, softbound, for $9.95, and published by *Datamost*, 20660 Nordhoff St., Chatsworth, CA 91311. The authors introduce the book with one very important note we'd like to pass on: "Try to solve each puzzle yourself, before looking up the answer." That's good advice, too, because you don't want to take the fun out of your game-playing, but "Shortcut" is good to have at the ready, if and when you need it. Betcha we put it to use soon!

And now the shadows grow long, a purple haze graces the Far Mountains, and our supper awaits us. We must shutter our windows, and be on our way.

For New Game Players

An Introduction to Eamon Adventures

Michael Thyng

Probably the first modern computer game that "everyone" heard about was *Pong*. But *Pong* is a game of coordination and wrist exercise. Likewise, so are *Pac-Man, Invaders,* and *Defender.*

The first computer game was probably some main-frame version of *Star Trek.* I keep running into various versions on the different systems I work with. Even in 1977, when the Apple] [was first introduced, sure enough, a cassette tape of *Apple] [Trek* running in Integer Basic, was included.

No one locally seems to know where the various versions originated. They just arrived, courtesy of someone who knew someone who knew someone who had a copy from someplace. This simulation of travel around the universe, fighting for the Federation, was no doubt the first computer adventure game. An adventure game is a computer program that describes a place by displaying words on the screen and gives the adventurer — you — a limited vocabulary of directions. The computer describes what surrounds you at a given stopping point, and you decide which direction to go. The object of the game is usually to recover the greatest number of valuable items in a dangerous place and return to the starting point without being killed. There are variations on this theme, such as rescuing unjustly imprisoned hostages or some important official.

One of the popular adventure games is called EAMON. Like the typical adventure game, playing EAMON is a quest to amass the greatest amount of loot and return to the main hall alive. If you've never played an adventure game, EAMON is a good one to get your first experience on. It offers not only an easy first adventure, The Beginners Cave, but it also gives you the programs necessary to create your own EAMON adventure, with your own creatures, places, and valuable items to be found.

To begin the game you just type "RUN THE WONDERFUL WORLD OF EAMON." You immediately find yourself in the main hall. A sign says "register here." The computer tells you that you see people drinking beer, then it asks you if you want to join the people drinking or register? Now you have to make a decision. Depending on your response, you will either be guided into an adventure or else be killed by a stab in the back, and be forced to start over. When you figure out how to get past the first obstacle, you are told what the attributes are for your character. (An attribute is a *quality*, such as wisdom, that — as in the case of EAMON — can be given to a character.)

Each character will be assigned a number from 1 to 25 for each attribute. The EAMON attributes are Agility, Hardiness, and Charisma. Generally, the higher the number in each attribute, the better chance your character will have for survival when meeting other characters in the adventure. A female character with a lot of charisma may be able to soften the heart of the hairy gorilla and avoid being killed. Someone with low agility may not be able to move quickly enough to keep from being injured by a hostile character.

In the EAMON series adventures, you are given a list of directives or code words you use to play the game. The directives are such things as NORTH, SOUTH, GET, DROP, and the like. These tell the computer that you want to go north or south or pick something up or set something down. As the adventure progresses, you may find yourself in a long north/south tunnel with a passageway to the east and a door to the west; there is a can of brown liquid on the ground. Now you have to decide what to do. What's behind the door? To find out, just enter a "W." If you say "get can" you will pick up the can, which could hold a potion that might save your life at some future point in the game (like after the hairy gorilla beats you up) — or the brown liquid it contains may be a poison for which there is no antidote. Experience pays.

Unlike typical adventure games, EAMON does not stand alone. There are 40 or more unique adventures, on as many diskettes. You can travel from adventure to adventure just by successfully [remaining alive] returning from an adventure.

Each is available through Public Domain software. Public Domain means the programs belong to everybody, and the author asks no royalties (payment) for his work. Public Domain diskettes can be copied and given to your friends without violating any laws. EAMON diskettes are available from *Call -A.P.P.L.E., The Apple Avocation Alliance,* and many Apple Computer user groups around the country.

The simple hints given here also, in general, apply to many of the other adventure games. If you haven't played adventure-type games before, here's your chance to get started at very little cost.

Reward!

- **Unique?**
- **Different?**
- **Unusual?**
- **Far Out?**

Whatever. Whatever kids are doing with their Apples, we want to know about. And so do you. You have the "connections"; we have the magazine.

For "tips" leading to publication of a story in *The Apple's Apprentice*, we will offer a free 12-month subscription or renewal.

Emerald City Publishing, Inc.
P.O. Box 582-AA
Santee, CA 92071

For New Apple Users:

Better Begin With BASIC

Mike Collins

This column is about programming in the BASIC language on an Apple][or Apple //e computer. Some of you may already be familiar with your Apple and may have already started to program on it, while others of you may have just gotten your computers recently, and haven't had a chance yet to play around on it. Or perhaps you have had an Apple at home or at school for a couple of years, but haven't wanted to do anything with it until now.

Everyone is different, but hopefully interested in learning to program in BASIC. That is why I decided to use this first column to try and teach you a little bit about the way we will be writing BASIC programs on our Apples.

First of all, it might help if we all know what a "program" really is. I looked program up in the dictionary, and got several answers, including *a brief outline of the order to be pursued or the subjects included*, and *coded instructions for a mechanism.* Both of these definitions are trying to say the same thing, but neither has made it very clear. As far as we are concerned, a program is *a sequence of instructions given to a computer to make it do something for us.*

For example, if you wanted to tell your Apple to type the word "HELLO" on the screen, we could write a program like this:

```
10  PRINT "HELLO"
20  END
```

That is a BASIC program that instructs or tells the computer to type the word "HELLO" on the screen where we can see it. As you can see it is pretty short, and very simple. When you have typed it in and then typed the *keyword* "RUN," it will *execute*, and you have written and run your first program!

> So now your dad (or mom) has bought a computer, and told you when they were not using it, you could play with it. What are you going to do, play games? Sure, that's a good idea, 'cause a couple came with with the machine, and mom and dad don't really dig games that much.
>
> But you know, after a few weeks, two or three games can get pretty dull. Then what? Well you Better Begin With BASIC. You already know all there is to know about DOS, or at least to put a disk in the drive and turn the power on, and then the same ol' silly game boots up.
>
> Mebbe you better stop and think: is there anything more exciting you can do? The Apple computer really is a pretty high-falutin' piece of equipment. Do you really have any idea of what it can do, of what you can make it do?
>
> Well, guys and gals, you Better Begin With BASIC. Applesoft Basic is a "programming language" that comes built into the Apple][. Mostly it is made up of simple words or instructions like PRINT or TEXT or IF or THEN. You'll be amazed how little study is needed to write a simple program that will print your name on the screen. If you're really interested in seeing how much fun you can have learning to program your Apple, Mike Collins has agreed to take the time to show you.

The program is only two lines long, and each line has a number. It is standard practice for most programmers to number the lines in their programs in steps of ten, just in case they want to add something inbetween lines later on without having to retype the whole program.

For example, if I wanted to say hello to my friend Danny on the Apple, I could use the program we just wrote to do it. If I type RUN, then the word HELLO would appear on the screen, just as I explained a moment ago. What if I wanted to put Danny's name on the screen at the same time? If I could get the computer to type HELLO DANNY, then that would be a much nicer message than a plain, ordinary HELLO. That way, Danny would know I was saying "hello" to him, and not just anybody that happened to come along and type RUN on my Apple.

I could change the program to say HELLO DANNY in a couple of different ways. First, I could just rewrite line 10 to read:

```
10  PRINT "HELLO DANNY"
```

That would do the trick, and when I ran the program by typing RUN, I would see the message HELLO DANNY just as I had expected. Suppose I was lazy and didn't want to retype the line. I could do the same thing by adding another line between 10 and 20, which would tell the computer to type Danny's name after it had typed HELLO. The way to do this is to pick a number between 10 and 20, like 15, and type in:

```
15  PRINT "DANNY"
```

If I want to see the whole program together, I can type LIST, and the screen will show:

```
10  PRINT"HELLO"
15  PRINT"DANNY"
20  END
```

LIST is the instruction that tells the Apple to show me what my program looks like so far, and when I type RUN, I will see:

```
HELLO
DANNY
```

That's just what I wanted to see. By adding in a line number that was between the line that told the Apple to print HELLO, but before the one which told the machine the program was over (the END statement), I was able to tell the computer to say hello to Danny, rather than just hello.

The most important thing to learn from the program is the process we had to go through to make the computer do what we wanted. In order to make the machine say "hello," we had to tell it to PRINT "HELLO". Then, to have the Apple say hello to Danny, we had to find the part of the program after it said HELLO, but before it said END, and tell it to PRINT "DANNY." If we had tried to do the same thing just by typing PRINT "DANNY" at the beginning of the program, when we tried to RUN it, we would see:

```
DANNY
HELLO
```

which doesn't make very much sense. In other words, to make the Apple do what we wanted, we had to tell it each special step involved in doing the task, *in the order we wanted.* This looks pretty easy, and it is, but sometimes there are more than just these few steps involved in running a program.

For example, when we take out the garbage, all we usually think about is taking out the garbage. We don't go through the chore piece by piece, telling ourselves to:

1) get the garbage
2) carry the garbage outside
3) put the garbage in the can
4) go back inside

However, if we wanted to have the computer do the job for us, that is exactly what we would have to do. We would have to write a program to tell the computer each and every step it needs to do to make sure the garbage got emptied O.K. The program might look something like this:

```
10 GET GARBAGE
20 CARRY GARBAGE OUTSIDE
30 PUT GARBAGE IN CAN
40 GO BACK INSIDE
```

Even this is an over–simplification, because we forgot things like taking the top off of the garbage can and putting it back on, but you get the idea, and if our Apple were a robot which could pick things up and move them around, then the program might work. The important thing is to understand how to take a task, like taking out the garbage, and break it down into a series of little, tiny jobs. Once we have done all of that, then we can write the program that tells the computer what to do.

We can practice breaking things down into jobs by making up pretend programs to do things that we do without ever thinking about what we're doing. If you wanted to go to the kitchen to get a glass of milk, you could make up a program that would *simulate* your actions, like:

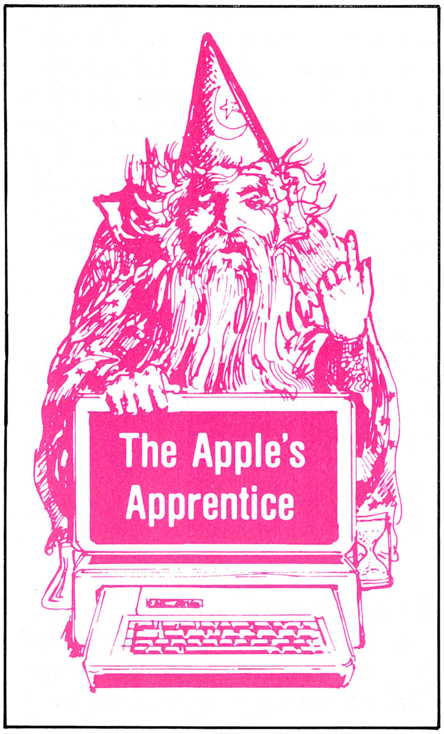
```
10   GO TO KITCHEN
20   OPEN REFRIGERATOR
30   GET MILK
40   CLOSE REFRIGERATOR
50   GET GLASS
60   TAKE TOP OFF BOTTLE
70   POUR MILK
80   PUT TOP ON BOTTLE
90   OPEN REFRIGERATOR
100  RETURN MILK
110  PICK UP MILK GLASS
120  DRINK MILK
130  GO TO SINK
140  TURN ON WATER
150  WASH GLASS
160  TURN OFF WATER
170  PUT GLASS IN CUPBOARD
180  LEAVE KITCHEN
```

When we write it out like that, we can see that there are a whole lot of different jobs involved in doing something simple like getting a glass of milk, so a computer must be told each and every step along the way. Even as detailed as we were, a real computer would have to be told even more than the instructions we put in out program. It would have to know where to get the milk glass, how to pour the milk, how to drink it, how to turn on the faucet to wash the glass, and a whole bunch of other things we never think about until we have to explain it to a computer.

Computer programmers have a saying: Garbage in, garbage out. This means that, because a computer can do only what you to it to do, if you tell it to do something that doesn't make any sense, then it will do something that doesn't make any sense. The key to programming your Apple is knowing exactly what you want your Apple to do for you, and working out exactly the way to tell it, so that it will understand what you mean.

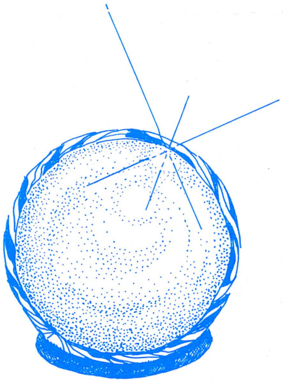
In future columns I will help you to learn about what words or commands your Apple understands, so that you will know how to tell it what to do. Remember, the Apple is your servant; you can use it for many things.

In the meantime, have fun. That's what it's all about!



New Products

The Crystal Ball



New Graphics System

Apollo Software Technology, a software research and develop company, recently published the Graph-Art System I, a graphic utility program for the Apple] [Plus and //e. The program can help users create brilliant pictures for entertainment, business display, or computer art hobbyist.

It includes a 50+ coloring program, a fast eraser, an inverse picture producer, a tracing sheet, a slide show organizer, mix graphic/text, and much more. The program is on a disk, with a tutorial/instruction booklet, and demo pictures (both on disk and booklet). The suggested retail price for the system is $34.00. For further information, please write to Apollo Software Technology, P.O. Box 34057, Houston, Texas 77234.

Hallowe'en Year 'Round

The fun and magic of pumpkins, spiders, and skulls is now available the year "round with 'HALLOWE'EN" from MicroSPARC. This high-resolution arcade game is for a 64k Apple] [, //e, or Franklin Ace.

The object of HALLOWE'EN is to lead a wizard through a 10-screen kingdom, snatching pumpkins and killing spiders and skulls along the way.

A novice wizard will fall prey to the pouncing spiders, bouncing skulls, and deadly choppers, unless he quickly becomes adept with his magic wand.

It can be played with one or two players, using joystick, paddles, or keyboard. Games can be redesigned extensively by varying options which include sound toggle, delay times for falling objects, choice of point values per object, and even the number of skulls, pumpkins, etc., on each of the 10 levels. The top five highest scores can be saved on disk.

HALLOWE'EN comes on a pumpkin-colored diskette and can be ordered for $29.95 from MicroSPARC, Inc., 10 Lewis St., Lincoln, MA 01773, phone (617) 259-9710. Check, money-order, Visa and MasterCard accepted.

M.E.C.C. Catalog

The current MECC catalog contains 190 educational computing programs and training materials for the Apple] [and other computers.

Request your free copy now by contacting Minnesota Educational Computing Consortium (MECC), 3490 Lexington Avenue North, St. Paul, MN 55112. (612) 481-3500.

AppleMouse] [Announced

Apple Computer demonstrated a mouse and new software called MousePaint, which is now available for its Apple®] [personal computers.

A mouse is a pointing device that can be used instead of a keyboard to select computer operations and modify information displayed on the screen. Apple's Lisa™ computer, introduced last year, ignited broad interest in the mouse among computer owners, software developers and other computer manufacturers. The Lisa is designed to be almost completely controlled by a mouse, one of the things that makes it much easier to use yet more powerful than conventional computers.

AppleMouse] [will be packaged with MousePaint software that uses the mouse to design charts, diagrams, free-hand drawings and other visual aids for reports and presentations. Users can insert text in a drawing and can choose from a variety of character fonts and fill patterns. MousePaint simulates bit map graphics to support the Apple] ['s high resolution capabilities.

Independent software developers are being encouraged to write applications that take advantage of the mouse. Most future Apple software programs for the Apple] [family will offer the mouse as an option.

Apple also is developing a mouse to work with Apple /// personal computers.

Apple Color Plotter

Apple Computer has introduced a multi-color plotter for its Apple®] [and Apple /// personal computers that produces presentation-quality graphs, charts, drawings and graphic designs on either paper or overhead transparencies.

The Apple Color Plotter, Model 410, is a versatile tool for anyone who gives presentations or needs to produce printed copies of multi-color graphics created with a personal computer.

The Apple Color Plotter is compatible with Apple Business Graphics, Apple Logo and many other software programs which utilize the high-resolution, multi-color graphics capabilities of the Apple] [and Apple ///.

The plotter features four color pens that are interchanged automatically during operation. Commands given to the plotter by the graphics software determine pen selection, so a user does not need to stop the plotter's operation to change pens.

Designed for both plotting accuracy and speed, the plotter provides 0.004-inch resolution, 0.008-inch repeatability and 3.94 inches-per-second plotting speed. Its variable-width plotting bed and adjustable pinch roller mechanism permit a wide choice of media sizes up to 11″ x 17″.

The Apple Color Plotter uses a standard RS-232C serial interface and will connect to any personal computer that supports this interface. The plotter connects directly to the Apple ///'s built-in serial port. Apple's Super Serial Interface Card connects it to the Apple] [Apple] [+, or Apple //e.

The Apple Color Plotter is available now from U.S. authorized Apple dealers and through Apple's educational state buy programs. The U.S. suggested retail price for the plotter is $995. The plotter carries Apple's standard 90-day warranty, and Apple's extended service plan, AppleCare™, also is available.

New Educational Software

The Learning Company, producer of award-winning educational software packages, has introduced five new discovery learning programs... Reader Rabbit and The Fabulous Word Factory, Word Spinner, Number Stumper, Addition Magician and Colorasaurus.

With these new offerings, The Learning Company product line has two new reading programs, two new number programs and its first art program.

Reading Programs

Reader Rabbit and The Fabulous Word Factory is a completely new approach to teaching essential pre-reading and early reading skills to children aged 5 to 7. By helping Reader Rabbit match pictures and words, label boxes and load words into a word train, children learn to recognize and spell more than 200 three-letter words. The program sells for $39.95.

Word Spinner takes young readers, aged 6 to 10, on a whirl through the alphabet to learn the building blocks of reading. In building more than 500 three-letter words and 1,000 four-letter words, children learn to recognize word patterns and develop critical vocabulary and spelling skills. Emphasis is placed on proper positioning of vowels and consonants to enhance word pattern recognition. Priced at $34.95, the program runs on Apple][computers and is now available.

Number Programs

Number Stumper, the computer-age version of a classic 14th century game of chance, teaches basic addition and subtraction. Using animated dice, children aged 6 to 10 create and solve mathematical equations and develop abstract reasoning and strategic thinking skills. The program sells for $39.95.

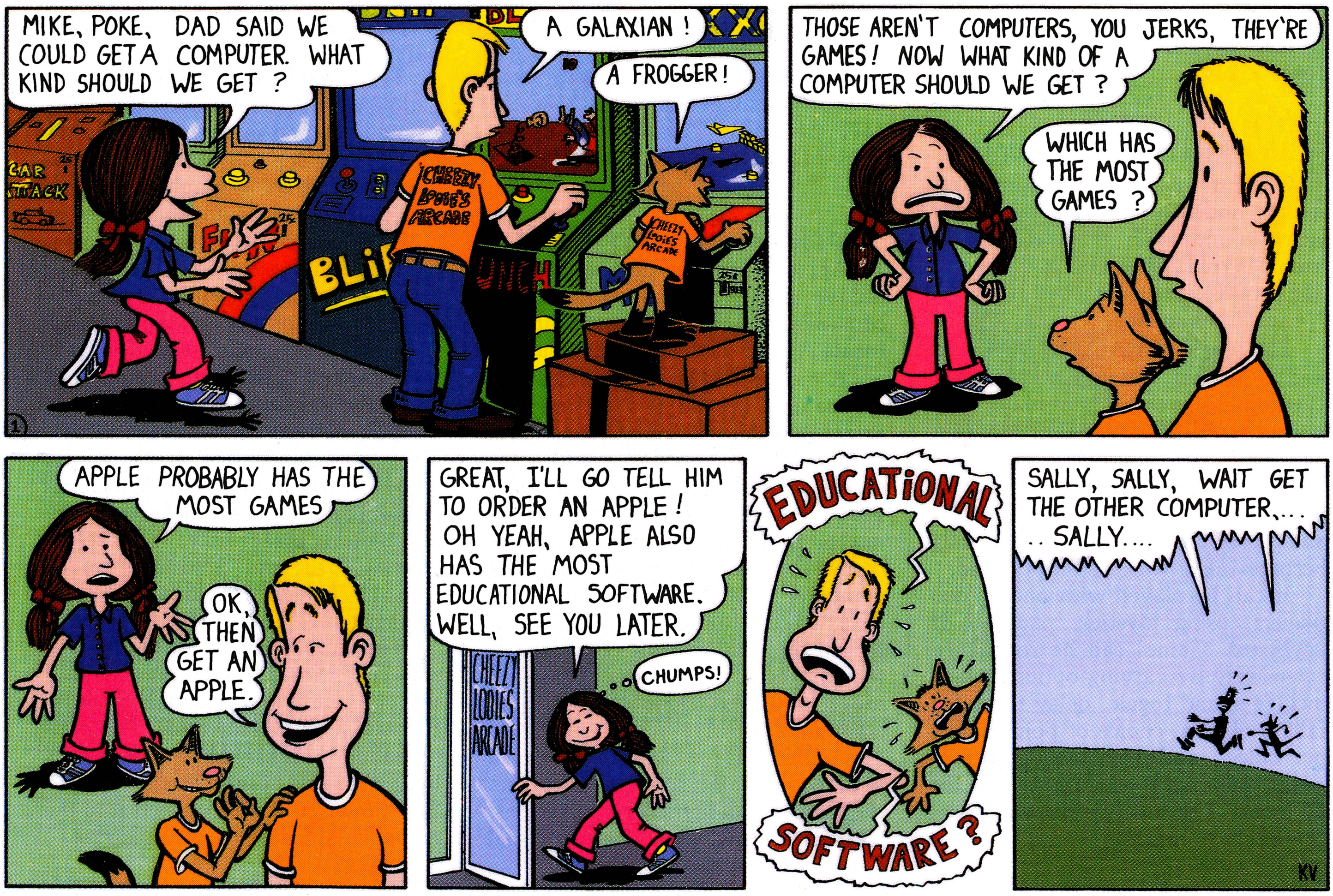
Addition Magician is a fast-paced, race-against-the-clock number strategy game that teaches children the basic concepts of addition and flexible thinking about numbers. Children build magic walls around groups of numbers that add up to a target sum. The program, for children aged 6 to 10, is priced at $34.95.

With these additions, The Learning Company now offers 14 educational software packages. Its other programs are Juggles' Rainbow, Bumble Games, Bumble Plot, Gertrude's Puzzles, Gertrude's Secrets, Moptown Parade, Moptown Hotel, Magic Spells and the popular Rocky's Boots, called a "software classic" by The New York Times.

The Learning Company is headquartered at 545 Middlefield Road, Suite 170, Menlo Park, CA 94025. Telephone (415) 328-5410.

MICROCHIPS

by Robert Cavey


MIKE, POKE, DAD SAID WE COULD GET A COMPUTER. WHAT KIND SHOULD WE GET ?
A GALAXIAN !
A FROGGER!
CAR ATTACK
BLI
CHEEZY LODIE'S ARCADE
THOSE AREN'T COMPUTERS, YOU JERKS, THEY'RE GAMES! NOW WHAT KIND OF A COMPUTER SHOULD WE GET ?
WHICH HAS THE MOST GAMES ?
APPLE PROBABLY HAS THE MOST GAMES
OK, THEN GET AN APPLE.
GREAT, I'LL GO TELL HIM TO ORDER AN APPLE! OH YEAH, APPLE ALSO HAS THE MOST EDUCATIONAL SOFTWARE. WELL, SEE YOU LATER.
CHUMPS!
CHEEZY LODIES ARCADE
EDUCATIONAL
SOFTWARE ?
SALLY, SALLY, WAIT GET THE OTHER COMPUTER.... .. SALLY....
KV

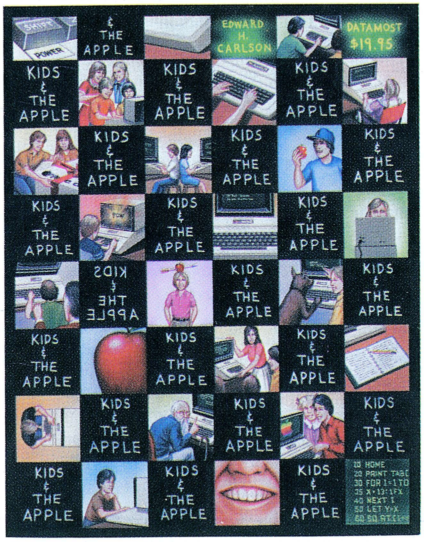
The Book

That's an Open Door to the Computer Future

KIDS & THE APPLE is its name, and its game is to prepare your child, or any child, to take his or her place as a member of the computer generation by teaching them the mysteries of the Apple* computer in ways they'll love and enjoy. Don't be surprised if you will also learn along with your child.

The kids of today are fascinated by computers to start with. And that's great, because it means **they're eager** to learn. But, until this book by Edward H. Carlson, learning about the Apple was a fumbling, bumbling effort for a child.

KIDS & THE APPLE was designed in every aspect to lead them gently, interestingly yet quickly into the computer world. First, it's a large 8½ by 11 book which can be opened flat for ease of use. Second, there are 35 chapters, each one building upon the knowledge of the prior chapter – and it's loaded with dozens and dozens of cartoons which make a point as they amuse.



At computer stores, or from:

The educational/book division of

DATAMOST INC

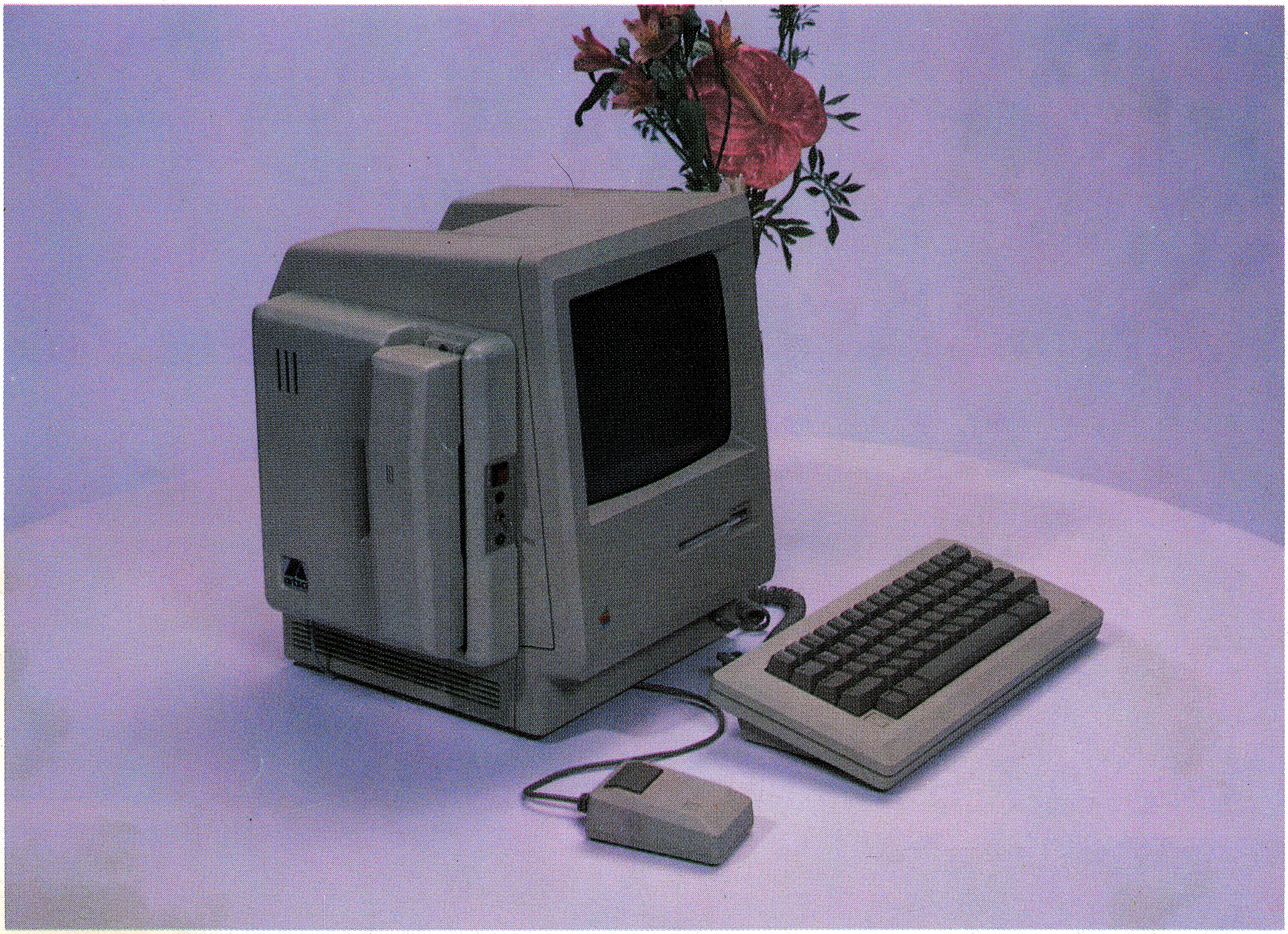
9748 Cozycroft Ave.
Chatsworth, CA 91311 (213) 709-1202

Then, there are special sections for a parent or teacher to use so they can work along with the kids, if they wish, and help them over any rough spots.

Perhaps the major reasons the kids will love this book is that it is **truly** written so they can easily understand it (without a lot of confusing technical language) . . . and that they see on-screen-results almost immediately! Right away they realize they'll soon be programming their Apple, making their own games! . . . or creating other programs for school or work or to play.

The computer world is roaring toward us. To be successful at work, school or even play, a child will have to be knowledgeable about computers. Make sure your favorite child is prepared for the challenge. With KIDS & THE APPLE at his side, he'll enjoy learning and you'll know you've prepared him or her for a successful future. Only $19.95.

*Apple is a trademark of Apple Computer, Inc.



The Macintosh has a new ring.

The ARTSCI **MAGICphone**™ creates an entirely new aspect to the APPLE Macintosh™, that provides a spectrum of voice/telephone communications.

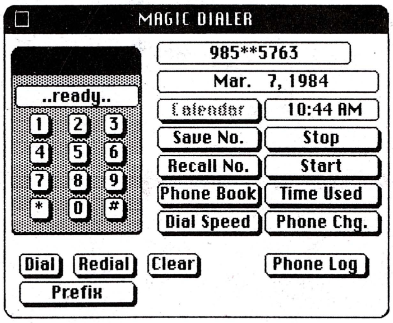
Software is provided to manage and dial hundreds of phone numbers and will record and print the details (time, number, and charge) of each call. The phone log can be printed on a daily, weekly, or monthly basis. The mouse is used to select and dial any phone number, and **MAGICphone** uses TONE dialing to any of the lower-cost telephone carriers like SPRINT and MCI.

The **MAGICphone** also allows control of two separate phone lines. A "hold" feature for each line with lighted indicators is provided. You can then call someone to the phone or pick up an extension and continue the call. The **MAGICphone** can temporarily stop incoming calls, and the caller will hear a busy signal.

The **MAGICphone** can be used as a stand alone telephone without Macintosh control. Tone dialing and last number redial are standard features.

Installation is a snap. You simply plug the **MAGICphone** into your Macintosh speaker jack, and into your phone line.

Available now. $199.95.



985**5763

Mar. 7, 1984

10:44 AM

ARTSCI, INC.
5547 Satsuma Avenue
North Hollywood, CA 91601
818/985-2922

MAGICphone was developed for ARTSCI by Digital Matrix Inc
MAGICphone is a trademark of Artsci, Inc.
Macintosh is a trademark of APPLE Computer, Inc

APP50APPLEAP
2⁵⁰